Power Management in Disruption Tolerant Networks

A Thesis
Presented to
The Academic Faculty

by

Hyewon Jun

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

College of Computing
Georgia Institute of Technology
December 2007

Power Management in Disruption Tolerant Networks

Approved by:

Professor Mostafa H. Ammar
(College of Computing), Advisor

Professor Mark Corner
(Computer Science, University of Massachussette)

Professor Umakishore Ramachandran
(College of Computing)

Professor Raghupathy Sivakumar
(School of Electrical and Computer Engineering)

Professor Ellen W. Zegura
(College of Computing), Advisor

Date Approved: 24 October 2007

To my husband Seung and my parents

for their love and support

ACKNOWLEDGEMENTS

First of all, I sincerely want to thank my advisors Dr. Mostafa Ammar and Dr. Ellen Zegura for their guidance, support and encouragement. They provided me the flexibility to choose projects while guiding me into the right directions. Also, they gave me constructive criticisms as well as rewarding praises, which motivated and trained me to improve the quality of my research. Furthermore, their insights and experiences were so valuable for me to enhance my research and presentation skills. Even short meetings with them resulted in significant improvement. In addition, their hospitality during my pregnancy and raising my son helped me to achieve my degree with comfort and confidence.

I also want to thank Dr. Mark Corner for his help and efforts during our collaboration. I learned a lot from his detail oriented attitude and dedication to complete the high quality research. As a result, the project I conducted with him and my advisors turned out to be my favorite project.

In addition, I want to thank the rest of my thesis committee members, Dr. Umakishore Ramachandran and Dr. Raghupathy SivaKumar, for their time and valuable feedback. Their comments helped me to consider various aspects of my research and to improve the dissertation in many ways.

I also want to acknowledge all the students and staffs in College of Computing, especially those in the Networking and Telecommunication Group for their support and friendship. The unique family-like community environment made me to survive the tough graduate student life with comfort.

Finally, I would like to thank my family. My parents and younger sister have always provided me with unconditional love and support. Without their faith in me, I could not have made it this far. I also want to thank my husband Seung. Meeting him at the beginning of my Ph.D. program made my life much enjoyable and comfortable. His continuous love and support helped me to have confidence on myself. He was also an excellent colleague who challenged my research in various ways and gave me critical comments. I also want to thank my parents-in-law for their support and pride in me. At last, I want to thank my son Jaywu for his lovely character. He is my delightful joy as well as another great achievement during my Ph.D. student life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Disruption Tolerant Networks (DTNs) are mobile wireless networks that are designed to work in highly-challenged environments where the density of nodes is insufficient to support direct end-to-end communication. Recent efforts in DTNs have shown that mobility provides a powerful means for delivering messages in such highly-challenging environments. Unfortunately, many mobility scenarios depend on untethered devices with limited energy supplies. Without careful management, depleted energy supplies will degrade network connectivity and counteract the robustness gained by mobility. A primary concern is the energy consumed by wireless communications because the wireless interface is one of the largest energy consumers in mobile devices whether they are actively communicating or just listening. However, mobile devices exhibit a tension between saving energy and providing connectivity through opportunistic encounters. In order to pass messages, the device must discover communication opportunities with other nodes. At the same time, energy can be conserved by "sleeping," i.e., turning off or disabling the wireless interfaces. However, if the wireless interface is asleep, the node cannot discover other nodes for communication. Thus, power management in DTNs must balance the discovery of other nodes while aggressively sleeping the radio during the remaining periods.

In this thesis, we first develop a power management framework for a single radio architecture that allows a node to save energy while discovering communication opportunities. The framework is tailored to the available knowledge about network connectivity over time. Further, the framework supports explicit trade-offs between energy savings and connectivity, so network operators can choose, for example, to conserve energy at the cost of reduced message delivery performance. We next examine the possibility of using a hierarchical radio architecture in which nodes are equipped with two complementary radios: a long-range, high-power radio and a short-range, low-power radio. In this

architecture, energy can be conserved by using the low-power radio to discover communication opportunities with other nodes and waking the high-power radio to undertake the data transmission. However, the short range of the low-power radio may result in missing communication opportunities. Thus, we develop a generalized power management framework in which both radios support the discovery. In addition, we incorporate the knowledge of traffic load and network dynamics and devise approximation algorithms to control the sleep/wake-up cycling of the radios to provide maximum energy conservation while discovering enough communication opportunities to handle the expected traffic load. Finally, we investigate the *Message Ferrying (MF)* routing paradigm as a means to save energy while trading off data delivery delay. In MF, special nodes called ferries move around the deployment area to deliver messages for nodes. While this routing paradigm has been developed mainly to deliver messages in partitioned networks, here we explore its use in a connected MANET. The reliance on the movement of the ferries to deliver messages increases the delivery delay if a network is not partitioned. However, delegating message delivery to the ferries provides the opportunity for nodes to save energy by aggressively putting their radios to sleep when ferries are far away. To exploit this feature, we present a power management framework, in which nodes switch their power management modes based on the knowledge of ferry location.

CHAPTER I

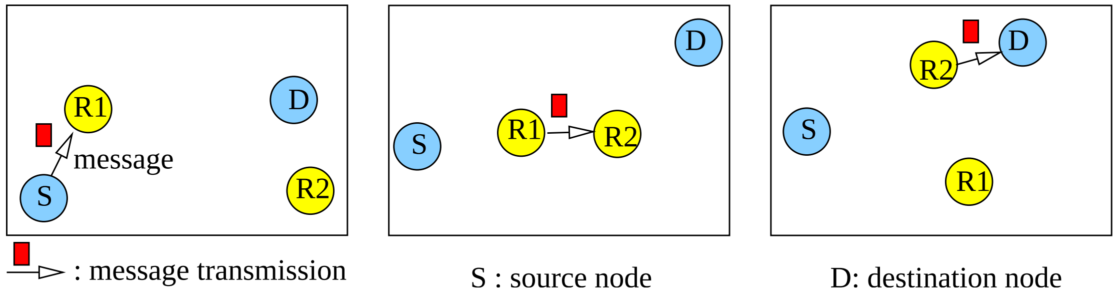
INTRODUCTION

Wireless networks proliferate people's lives from portable devices such as laptops and iPAQs to mobile devices such as cellphones. People demand more data services while they are moving. In fact, 3G data services and WiMAX infrastructures are under development to cooperate such a demand. In addition, we are moving toward the era of ubiquitous computing, in which information processing is thoroughly integrated into everyday lives through small, inexpensive, networked devices. So, people have more freedom to move while information is processed in networked environments. Thus, wireless technologies keep providing great amount of flexibility to everyday human life.

Recently, many research efforts have been devoted to extending such flexibility to challenged environments such as deep space, disaster-relive sites, highway, sensor fields, battle fields, social events, and rural areas by providing wireless networking services [19, 6, 8, 54, 10, 9, 7, 22, 29, 35, 62, 67, 51]. Networking in these environments has important applications such as communication services for scientific data delivery, navigator services for the explorer in hostile environments, temporary communication services where no infrastructure exists, computation services in motion (e.g., to find a map), and monitoring services for inaccessible and hostile environments. However, building such network services faces new odds such as disruptions on their connections due to mobility, geographical obstacles, and low density of nodes. In traditional mobile ad hoc networks (MANETs), nodes are deployed dense enough to provide end-to-end paths among nodes and node mobility is assumed to be low. While this approach provides a way to deliver data fast, deploying a dense network could be unnecessarily expensive in challenged environments. For example, when observing zebras' group behavior, deploying sensors densely in the whole national park could be expensive. As an alternative, in ZebraNet [35], sensors are attached to zebras and vehicles loaded with base stations move nearby the group of zebras to collect information from the sensors from time to time. Collecting data in this alternative network scenario achieves equivalent performance to that of the dense deployment scenario at much lower cost, while increasing latency. Besides the economic reason, some environments do not allow the MANETs approach at all. In military ad-hoc networks, deploying dense wireless network in the entire battle field is not feasible. Therefore, exploring the characteristics of the challenged environments and developing architectures to build network services in disruptive environments is important to support aforementioned applications.

A Disruption Tolerant Network (DTN) is a generalized networking architecture to build network services in such challenged environments with connection disruptions [25]. The disruptions in DTNs occur due to many factors such as node mobility, physical obstacles, depleted energy, and low node density. As a result, a DTN becomes partitioned in various ways and nodes may not have end-to-end paths among them at any point of time. In such an environment, data can be delivered by mobile nodes which *store, carry*, and then *forward* data toward destinations [23, 30, 18]. Figure 1 shows an example scenario. Initially, node S has a message to node D. However, node D is not within the radio range of node S. As time passes, nodes move around in the deployment area. In Figure 1(a), node S meets node R_1 within its radio range, so it forwards the message to node R_1. Then, node R_1 stores the message, carries it until encountering node R_2 as in Figure 1(b) and then forwards it. Finally, node R_2 carries it until meeting node D as in Figure 1(c) and then forwards it to the destination, node D. This delivery paradigm introduces long delivery delay because the physical movement of nodes takes relatively long time compared to radio transmission delay. Because of these characteristics, interactive protocols such as TCP do not work in DTNs, and new protocol stacks need to be developed for DTNs.

One of the major challenges in building the DTN architecture is routing protocol. As illustrated in Figure 1, nodes have communication opportunities (also called *contacts*) based on their radio ranges and physical locations along time. A series of such contacts among



(a) From node S to R1 (b) From node R1 to R2 (c) From node R2 to D

Figure 1: An example scenario of message forwarding in a DTN as time passes

nodes provides a routing path from a source toward a destination. This route can be completely predictable if nodes move on fixed schedules, while it can be completely unpredictable (i.e., opportunistic) if nodes move in an arbitrary manner. Many studies have identified the characteristics of a set of DTNs and proposed routing protocols on them. Such routing protocols can be classified into three categories: *knowledge-based mechanisms*, *designated delivery mechanisms*, and *opportunistic mechanisms*. First, knowledge-based mechanisms utilize available knowledge about network dynamics and apply Dijkstra's algorithm to find the shortest path in terms of expected delay [30, 18]. Second, designated delivery mechanisms designate special mobile nodes to deliver messages among other nodes as in Message Ferrying [78, 79, 80] and Data Mule [62]. Finally, opportunistic mechanisms forward one or more copy of messages to other nodes without any knowledge about network dynamics and hope for eventual delivery [72, 69, 73, 31].

Another major issue in DTNs is energy conservation. In the class of DTNs, there are important applications with energy constraints. For example, nodes deployed in a remote or hazardous area may have limited access to energy sources, yet the need for long network lifetime. Even if some nodes have abundant energy, a heterogeneous network may include key devices with limited energy. Thus, efficient power management mechanisms are necessary to allow these networks to remain operational over a long period of time. However, mobile devices exhibit a tension between saving energy and providing connectivity. In order to pass messages, the device must discover other nodes, typically using the same wireless interface used for message transfer. At the same time, energy can be conserved by disabling or turning off (i.e., sleeping) the wireless interface because the wireless interface is one of the largest energy consumers in mobile devices whether they are actively communicating or just listening [45, 26, 70]. However, if the wireless interface is asleep to save energy, the node cannot communicate with other nodes.

Power management in traditional MANETs have explored this feature of wireless interfaces also. However, the main assumption in the MANETs is that nodes are deployed densely enough to form a connected network all the time unless their radios are asleep. Thus, many power management mechanisms have been developed to save energy while keeping a network connected in some ways. That is, in 802.11 Power Saving Mode (PSM) [1], nodes wake up at synchronized times to form a connected network. When they are awake, they notify any pending messages to intended receivers, so the receivers stay awake and receive the messages. In another mechanism [71, 82], neighboring nodes wake up in such a way that their awake time intervals overlap one another, so that they connect a network eventually without the aid of clock synchronization among nodes. Also, when the node density is high, studies [20, 74] proposed to divide a development area into small cells and elect a few coordinators in each cell to form a connected network while others sleep. Finally, when an additional secondary low-power radio is affordable, nodes may utilize the low-power radio to keep a network connected and wake up the main communication radio only if needed [65, 64]. These approaches assume that a network is densely deployed, in which a node has another node within its radio range most of the time. Thus, they focus on how to overlap the time intervals in which the main communication radios are awake to form a connected network, while allowing the radios to sleep. However, in DTNs, a network may become partitioned for a long time. Thus, even when all nodes are awake, the network may not have any connection among nodes. Therefore, power management mechanisms in MANETs cannot be reused in DTNs.

The major issues in designing power management for DTNs is obvious to state, though challenging to realize in most practical environments: a node needs to determine when it can communicate with other nodes, so that it can sleep for the rest of the time. The time interval when two nodes can communicate with each other is called a *contact* [25, 30]. When networks are partitioned most of the time, it is not trivial to discover contacts while also saving energy. Specifically, the problems are classed into two questions: (1) *how to discover a contact* and (2) *when to search for contacts*. The discovery of another nodes can be done through various means such as radios, RFIDs, Infrared's, Ultrasonic or Sonic devices, and cameras. Depending on applications and situations, some devices are more appropriate than others. However, most of them use energy to discover other nodes. Therefore, the second question arises for most of discovery mechanisms. The question about when to search for contacts involves various factors such as any knowledge about node mobility and mobility characteristics. Nodes may know the exact time to have contacts with other nodes if their movement is precisely scheduled in advance, statistical knowledge from the past history, or no knowledge at all. Also, nodes may know the trajectory of other nodes, so it can estimate their possible contact time. The knowledge of such temporal or spatial information could be used to predict the node mobility and future contact time. If so, nodes can put their radios to sleep when they don't expect to have any contact for energy savings.

In this thesis, we consider to use a radio to discover contacts. We also examine the possibility of using a hierarchical radio architecture in which nodes are equipped with two complementary radios: a long-range, high-power radio and a short-range, low-power radio. As a system model, we assume an idealized system that has one platform with one or more wireless interfaces. For energy, we only account for the communication energy consumption of wireless interfaces and do not consider other sources such as computation or mobility. To address the energy conservation issue in DTNs, we develop a power management framework that allows a node to save energy while discovering contacts. Further, the framework supports explicit trade-offs between energy savings and connectivity, so network operators can choose, for example, to conserve energy at the cost of reduced message delivery performance. In the basic framework, a radio transits between three power management modes: a *dormant mode* (when no contact is expected), a *search mode* (when there is some expectation of a contact), and a *contact mode* (once a contact is established). Although the details of power management in each mode depends on the available capability of devices and criteria, the energy consumption in each mode decreases in the order of the contact, search, and dormant modes. As a result, the policy that determines when to transit among the power management modes determines the energy consumption of nodes. Also, the same policy addresses when to search for contacts, i.e., when to transit into the search mode or out of the search mode.

To determine when to search for contacts, we utilize available knowledge about network topology changes for three mobility categories: *regular mobility* (where contacts occur with a certain degree of regularity), *random mobility* (where contacts occur at random time), and *designated delivery* (where a designated mobile node is in charge of message delivery, so other nodes need to discover contacts only with the designated node). In the regular mobility case, we develop mechanisms for a single radio to search for contacts with different levels of knowledge. When contacts occur regularly, nodes have narrow time windows in which they can discover contacts with high probability. Thus, we devise algorithms to estimate the time windows with two forms of summary information about contacts (mean of time between contacts and contact duration, or mean+variance of time between contacts and contact duration). We also develop mechanisms for two extremes of knowledge (complete knowledge and no knowledge of contacts) to establish the basic ideas and comparison points.

In the random mobility case, nodes do not have such narrow time windows to discover contacts with high probability and may have next contacts at any time. So, nodes may need to search for contacts all the time. In this case, we investigate the usage of one additional low-power radio to support contact discovery in a hierarchical radio architecture, in which nodes are equipped with two complementary radios: a long-range, high-power radio and a short-range, low-power radio. In this architecture, energy can be conserved by using the low-power radio to discover contacts with other nodes and then waking up the high-power radio to undertake the data transmission. However, DTNs are generally applicable to sparse networks, where the low-power radio may reach a subset of other nodes that could be reached by the high-power radio. Therefore, if a node relies only on the low-power radio to discover contacts, it may miss them due to the shorter range. To avoid missing contacts, we propose a generalized power management scheme that uses both radios to participate in contact discovery. In addition, we incorporate knowledge about contacts and traffic load in the network and devise approximation algorithms to determine optimal parameters to minimize the overall energy consumption, while discovering enough contacts to handle the expected traffic load.

In the designated delivery case, a special node called *ferry* moves around the deployment area to deliver messages for other nodes. To investigate ideal and practical movement of the ferry, we assume that the ferry moves on a fixed route with either a *strict* schedule or a *loose* schedule in which nodes know the route. With a strict schedule, the ferry arrives at each location as it is scheduled. Thus, nodes can estimate when to meet the ferry precisely. With a loose schedule, the ferry is allowed to slow down or pause, which makes it hard to predict the ferry arrival at each location. In these scenarios, we use the spatial information about the ferry route and the temporal information about the ferry schedule to determine when to search for contacts to save energy without missing any contacts for mobile nodes as well as stationary nodes.

Besides designing the power management mechanisms in DTNs, we also ask a more fundamental question: *Is the DTN approach only a back-up approach to resolve networking issues when traditional mechanisms do not work?* We investigate how to use a DTN approach proactively for energy savings even when the traditional approaches work. Specifically, we consider a designated delivery case, Message Ferrying (MF), in a network with densely deployed nodes and study how to achieve energy savings by trading off latency. The use of MF can increase data delivery delay over traditional MANET multihop routing protocols (e.g., DSR [34], AODV [56], and DSDV [57]). However, it has important features that enable the network to save energy compared to these multihop routing approaches. First, utilizing the knowledge of ferry location, nodes can sleep without degrading performance when the ferry is out of communication range. Second, the ferry is in charge of data delivery, so nodes do not need to wake up to form a connected network because the ferry mobility eventually connects the network. Also, topology changes in MF do not result in any overhead to reconstruct routing tables. Finally, the movement of the ferry allows nodes to transmit data at different times according to their locations and decreases contention among nodes.

In this thesis, we explore the aforementioned issues to design power management mechanisms for DTNs. We first develop a power management framework to use a single radio for contact discovery and provide explicit parameters to balance between energy savings and performance. Second, we examine the use of an additional radio to support energy efficient contact discovery. In addition, we utilize the available knowledge about contacts and traffic load in the network to optimize power management. Finally, a designated delivery approach is investigated to provide energy efficient network services in a densely deployed network. In the following, we briefly describe these main components of this thesis.

1.1 A Framework and Knowledge-Based Mechanisms for a Single Radio Architecture

In this work, we leverage the observation that many DTNs are characterized by sparse connectivity, providing the opportunity to save energy by aggressively disabling node radios (sleeping) [39, 40]. The major challenge is to balance sleeping periods with wake-up periods so that valuable and infrequent communication opportunities between nodes are not missed. We first consider when a node has one common radio to discover other nodes and to communicate with them. In this platform, we develop a power management framework that allows a node to save energy while missing few communication opportunities. The framework is tailored to the available knowledge about network connectivity over time. Specifically, we develop mechanisms for two knowledge extremes: complete knowledge and no knowledge of contacts. These results establish the basic ideas and provide a context for the general and potentially more realistic case of partial knowledge. Further, the framework supports an explicit trade-off between energy savings and connectivity, so that network operators can choose, for example, to conserve energy at the cost of reduced message delivery performance. We evaluate our mechanisms using ns-2 simulations. Our results show that our power management mechanisms reduce energy consumption from 60% to 93% compared to the case without power management. These energy savings come at the cost of some performance degradation when available knowledge is incomplete, though our tuning parameter allows a rich set of tradeoffs between energy consumption and performance.

1.2 Hierarchical Power Management

In this work, we examine the possibility of using a hierarchical radio architecture, in which nodes are equipped with two complementary radios: a long-range, high-power radio and a short-range, low-power radio [36, 37, 38]. In this architecture, energy can be conserved by using the low-power radio to discover contacts with other nodes and waking the high-power radio to undertake the data transmission. However, DTNs are sparse networks, where the short-range radio may reach a subset of other nodes that could be reached by the high-power radio, even when the nodes are mobile. Therefore, if a node relies only on the low-power radio to discover contacts, it may miss them due to the shorter range. To avoid missing contacts, we propose a generalized power management mechanism that uses *both* radios to participate in contact discovery. This generalized scheme controls the *wake-up* intervals of the two radios, which it uses to trade between energy savings and the performance of message delivery. In addition, we devise an adaptive algorithm that decides *how* to sleep (i.e., turn off, disable, or stay on) based on the expected overhead for a given wake-up interval. We compare the generalized scheme with two alternative schemes: one that uses only the high-power radio for discovery, and one that uses only the low-power radio for discovery. We use ns-2 simulations to evaluate the schemes. Our results show that the scheme relying only on the low-power radio achieves the best energy efficiency in discovering contacts, while it may miss some contacts due to the shorter range. On the other hand, our generalized two-radio scheme can tune wake-up intervals of both radios to balance between energy efficiency and contact discovery performance. However, these gains heavily depend on what the wake-up interval of the radio is set to.

1.3 Traffic-Aware Optimization in Hierarchical Power Management

In the hierarchical power management architecture, wake-up intervals of each radio affect the energy savings and contact discovery greatly. However, the fact does not answer the question as to how to set the intervals in the first place. In this work, we incorporate the knowledge of traffic load and network topology changes and devise approximation algorithms to control the wake-up intervals (i.e., sleep/wake-up cycling) of the radios [36, 37, 38]. When the traffic load can be estimated, nodes do not need to discover all the contacts. They may discover just enough contacts to deliver the traffic load. Thus, we devise approximation algorithms to provide the maximum energy conservation while discovering enough communication opportunities to handle that load. We evaluate our schemes through simulations and compare them against single radio architectures, and against mechanisms that do not incorporate information about the load. The simulation results show that our approximation algorithm could reduce energy consumption from 60% to 99% compared with the case without power management. In addition, by evaluating power management schiemes in three different mobility scenarios, we show that the relative energy efficiency of using the additional low-power radio increases as the sparseness of the network increases.

1.4 Trading Latency for Energy using Message Ferrying

The design of power management is affected by routing protocols in use. We investigate a DTN routing paradigm, *Message Ferrying (MF)*, to save energy while trading off data delivery delay [41, 42, 43]. In MF, special nodes called *ferries* move around the deployment area to deliver messages for nodes. While this routing paradigm has been developed mainly to deliver messages in partitioned networks, here we explore its use in a connected MANET. The reliance on the movement of ferries to deliver messages increases the delivery delay if a network is not partitioned. However, delegating message delivery to ferries provides the opportunity for nodes to save energy by aggressively disabling their radios when ferries are far away. To exploit this feature, we present power management framework, in which nodes switch their power management modes based on knowledge of ferry location.

We evaluate the performance of our scheme using ns-2 simulations and compare it with Dynamic Source Routing (DSR). Our simulation results show that MF achieves energy savings as high as 95% compared to DSR without power management and still delivers more than 98% of data. In contrast, power-managed DSR delivers much less data than MF to achieve similar energy savings. In the scenario of heavy traffic load, power-managed DSR delivers less than 20% of data. MF also shows robust performance for highly mobile nodes, while the performance of DSR suffers significantly. Thus, delay tolerant applications should use MF rather than a multihop routing protocol to save energy efficiently when both routing approaches are available.

1.5 Outline

The remainder of this these is organized as follows. Chapter 2 summarizes related work. In Chapter 3, a power management framework and knowledge based mechanisms are developed. Chapter 4 designs hierarchical power management mechanisms and Chapter 5 optimizes the hierarchical power management using traffic load information in networks. In Chapter 6, MF is investigated as a means to trade latency for energy in a densely deployed network. Finally, Chapter 7 summarizes the contributions of this thesis and discuss the future work.

CHAPTER II

RELATED WORK

This chapter provides an overview of related work in DTNs and power management in wireless networks.

2.1 *Disruption/Delay Tolerant Networks*

A Disruption/Delay Tolerant Network (DTN) is a networking architecture to build network services in challenged environments [25]. In such an environment, connections between nodes are disrupted due to many factors such as node mobility, physical obstacles, depleted energy, and low node density. Thus, a DTN becomes partitioned in various ways over time and nodes may not have end-to-end paths among them at any point of time. Many research efforts have been devoted to develop robust network architectures in such challenged environments, and a common approach is to use mobile nodes to deliver data by *storing*, *carrying*, and then *forwarding* data toward destinations. For example, ZebraNet [35] tracks wildlife by attaching sensor nodes to animals and collecting information from the sensors using mobile base stations loaded on vehicles. Jain at el. [62] also utilize mobile nodes such as wireless devices carried by pedestrians to collect information from stationary sensor nodes. In DakNet [29], public transportation is used to provide broadband connectivity in rural areas in which an always-on infrastructure is not available. The Interplanetary Internet project [19] is conceived to utilize planets, satellites, and spacecrafts for the development of an Internet architecture in the deep space that provides communication services and navigation services for the explorer spacecrafts and orbiters of the future deep space missions. In Message Ferrying [78, 79, 80], special nodes called *ferries* proactively move around to deliver messages among mobile nodes as well as stationary nodes. Each network utilizes the characteristics of its environment and develop its own protocol stacks. Thus, it would be beneficial if we classify the characteristics of such challenged environments and provide guidelines and protocols to start with. In this section, we describe research efforts to provide a generalized architecture to support interoperability between various types of DTNs and routing protocols within each DTN.

2.1.1 A Generalized Architecture

To support interoperability between DTNs, Fall [25] proposes a generalized overlay architecture based on an asynchronous message switching paradigm. To address the unique characteristics of DTNs, this architecture has the following components.

- Gateway: To achieve interoperability between DTNs, special DTN gateways are located at the interconnection points of DTNs. These gateways are responsible for storing messages in nonvolatile storage if reliable delivery is required and translating protocols from one DTN to others.

- Late binding: To route messages among different DTNs, a name tuple is used to address locations. Each name tuple consists of two variable length parts. The first part is a globally unique, hierarchically structured region name. The second part identifies a name resolvable within the specific network. As a message is routed, only the first part is used until it reaches the edge of the destination DTN. Then, the second part is resolved into a local address.

- Custody transfer: To provide reliable delivery in DTNs with potentially high-loss rates, nodes delegate responsibility (called *custody*) for delivering messages to nodes with persistent storage space. This concept relieves potentially resource-poor end nodes from responsibility related to maintaining end-to-end connection state. Also, it is a generalized concept of end-to-end reliability.

- Class of service: Challenged networks often have limited resources. The class of service similar to a postal class of service provides a means for each DTN to allocate its buffer space, link capacity, processing time and power. This class can also be used for congestion control. However, how to allocate resources remains as future work.

- Path selection and scheduling: Because DTNs are often partitioned within or among them, a message route consists of a cascade of time dependent communication opportunities (i.e., contacts) among nodes. The predictability of contacts helps to choose next-hop nodes for messages as well as to select the next message to be sent. However, the details of path selection and message scheduling depends on the characteristics of each DTN and its routing algorithms.

While the detailed mechanisms of each component requires significant investigation, this generalized architecture emphasizes several design decisions worthy of consideration.

2.1.2 Routing Protocols

One of the major challenges in DTNs is the selection of routing paths. As illustrated before, nodes have different contacts based on their radio ranges and physical locations along time. A series of such contacts among nodes provides a routing path from a source toward a destination. Many studies have identified the characteristics of a set of DTNs and proposed routing protocols on them. In this section, we describe the overview of the related routing protocols in three categories: *knowledge-based mechanisms*, *designated delivery mechanisms*, and *opportunistic mechanisms*.

In knowledge-based mechanisms, nodes utilize various level of knowledge about network dynamics to select routing paths [30, 18]. Jain et al. [30] assume the existence of oracles that provide knowledge about when contacts occur (either exactly or statistically), how much buffer is occupied in each node, or traffic demand at any time. They present different routing algorithms based on how much knowledge is available in a spectrum from no knowledge to complete knowledge. When no knowledge is available, a message is forwarded to the first available contact. When various level of partial knowledge is available, they estimate expected delay to wait for a contact, to be transmitted, and to reach the next-hop node on each edge between nodes. They assign the aggregated delay to a weight on the edge and use modified Dijkstra's algorithm to find the shortest paths in terms of delivery delay. When the complete knowledge is available through all the oracles, they present a Linear Programming formulation to find the shortest paths in terms of delivery delay. However, in practice, such oracles may not be available unless node movement is precisely scheduled in advance. Thus, in MaxProp, each node keeps track of the probability of meeting peer nodes and exchanges that information when it encounters its peers. The probability serves as a weight on the edge between nodes to find the shortest path through Dijkstra's algorithm.

In designated delivery mechanisms, special mobile nodes are in charge of delivering messages among other nodes [78, 79, 80, 62, 29]. Zhao et al. [78, 79, 80] propose Message Ferrying, in which the special nodes proactively move around to deliver messages among stationary nodes as well as mobile nodes. In various scenarios, they design the node movement for the special nodes to support network services. In Data Mule [62], Jain et al. present an analytical model to understand the relationship between performance metrics and system parameters in such mechanisms. In DakNet [29], public transportation is used to provide broadband connectivity in rural areas in which an always-on infrastructure is not available.

Finally, in opportunistic mechanisms, one or more copies of messages are forwarded to other nodes, hoping for eventual delivery [72, 69, 73, 31]. In Epidemic routing [72], Vahdat et al. propose a flooding-based mechanism in which two nodes exchange their storage information to determine which messages stored in the peer node have not been seen by the local node. Then, each node requests copies of messages that it has not yet seen. This mechanism maximizes the delivery rate and minimizes the delivery delay. However, the overhead in the network is significant. Thus, Spyropoulos et al. propose to limit the maximum number of duplicated messages without much performance degradation [69]. Also, instead of simple duplication, other studies [73, 31] suggest to use erasure coding mechanisms to add redundancy in messages while reducing the overhead in a network.

2.2 *Power Management in Wireless Networks*

In energy-limited devices, the wireless interface is one of the largest consumers of energy [45]. The wireless interface consumes energy not only while actively communicating, but also while listening for data. In addition, the energy consumed by the wireless interface depends on the radio range and data rate. Longer radio range and higher data rate require more energy consumption than the others. In this section, we review related power management approaches in wireless networks.

2.2.1 Sleep/wake-up Approaches

Measurement studies show that the energy consumption of a wireless interface while listening to data is almost as high as that while actually receiving data [26, 70]. Thus, nodes can save considerable energy by "sleeping," i.e., turning off or disabling their radios, if not used. The main issue is how to balance the sleeping period and wake-up period, so the network performance does not degrade. In this section, we describe the overview of power management mechanisms investigating this issue in three categories: cycling between sleeping and waking up, designing energy efficient MAC protocols in dense networks, and sleeping based on traffic pattern.

In traditional multihop ad hoc networks (MANETs), many sleep/wake-up cycling mechanisms have been developed to save energy while keeping network connectivity. They fall into four classes: *synchronous*, *asynchronous*, *cell-based*, and *on-demand mechanisms*. In synchronous mechanisms such as 802.11 Power Saving Mode (PSM) [1], nodes wake up at synchronized times and notify pending messages to intended receivers in order to make them stay awake. In asynchronous mechanisms [71, 82], neighboring nodes wake up in such a way that their awake time intervals overlap one another, so that they form a connected network eventually without the aid of clock synchronization among nodes. In this approach, the shortage of time information forces nodes to stay awake longer than in synchronous mechanisms. In cell-based mechanisms [20, 74], the deployment area is divided into cells, and a few coordinators in each cell are elected to connect the network while others sleep. This approach utilizes the spatial redundancy in densely deployed networks. Finally, in on-demand approaches [65, 64], nodes are assumed to have secondary low-power channels to connect a network and wake up the main communication channel, if needed. These approaches assume that a network is densely deployed, in which a node has another node within its radio range most of the time. Thus, they focus on how to overlap the time intervals in which nodes are awake to form a connected network, while allowing nodes to sleep.

Energy efficient Medium Access Control (MAC) protocols are also developed to utilize sleeping and waking up of radios for energy savings[66, 75, 58, 21, 65]. The common approach is that a node sleeps while the wireless medium is busy due to any transmission within its radio range. These studies are initiated by observing that transmission of multiple nodes in wireless medium interfere with each other, so at most two nodes can communicate within a radio range at a time. Thus, if a node sleeps while others are communicating, it can save energy without sacrificing its throughput. Therefore, they propose mechanisms to increase sleeping time based on traffic activity in the neighborhood. As a consequence, they are useful in a dense network rather than in a sparse network like a DTN.

Traffic patterns are also important factors in the design of power management mechanisms. Zheng et al. [83] presented on-demand power management, in which a node switches its power management modes between Continuous Aware Mode and Power Saving Mode according to incoming data traffic in a wireless LAN using 802.11 MAC protocol. They observe that once a packet arrives, more packets tend to follow and form a flow. Thus, if a node receives a packet, it stays awake to increase throughput and decrease latency for the duration of a flow. Anand et al. [11] also proposed to tune power management modes adaptively to the application and network characteristics.

2.2.2 Hierarchical Power Management

Hierarchical Power Management utilizes multiple layers of radios to assist communication while saving energy. The radios are different in terms of power usage, data rate, or radio range. Often low-power radios have lower data rate as well as shorter radio range than the main radio. The Hierarchical Power Management mechanisms have been implemented in various forms [64, 68, 55]. Each utilizes low-power radios to listen for incoming signals and to wake up main devices, if needed. Also, studies in [15, 5, 12] propose to save energy by offloading data traffic in a low rate to a low-power radio or to increase the capacity of a mesh network. However, all of them assume that both radios are within range of other nodes. So, they did not investigate the impact of the short radio range of the low-power devices to networking performance in sparse networks.

In sparse ad-hoc networks, Banerjee et al. also use an additional radio for contact discovery, but consider the use of a high-power extra-long-range radio, which has longer range than, but similar power usage to 802.11 wireless cards [16]. To maximize contact time while saving energy, the extra-long-range radio listens for signals and wake up the main radio if it expects any node within its communication radio range of the main radio. Also, the extra-long-range radio cycles between sleeping and waking up with a long duty cycle. Due to the limitation on node speed, the extra-long range radio does not need to listen all the time to determine when a peer node arrive within its communication range of the main radio. Thus, with a long duty cycle, energy can be saved. The authors evaluate their scheme on a special case of mobility scenario to predict when any node will come in contact with a stationary node. However, the overhead of radios make it questionable whether the extra-long-range radio is more efficient than low-power, short-range radios in various mobility scenarios.

2.2.3 Energy Conservation using Node Mobility

In sparse networks, nodes can exchange messages by increasing transmission power to form a connected network with long radio range. However, the energy required to increase the transmission range is the multiple power of the range [60]. Thus, increasing the transmission power is not energy efficient. On the other hand, recent network design efforts have presented to use mobile nodes to deliver messages [62, 78, 29]. These studies use special mobile nodes to receive messages from source nodes, travel, and then forward to destination nodes. They focus on not increasing transmission power on their studies, without designing specific power management in their networks. They may take more advantage of the mobile nodes if proper power management mechanisms are designed.

Also, Goldenberg et al. propose to control node mobility to optimize the network performance such as energy efficiency based on specific traffic demands while using multihop routing protocols [28]. This approach requires nodes with extra capability to control their movement.

2.2.4 Energy and Latency Trade-offs

Trading latency for energy has been investigated based on the modulation scaling theory. According to the theory, the transmission power is proportional to the transmission rate. Therefore, sending a packet slowly using low transmission power can save energy. However, packets cannot be transmitted arbitrarily slowly on a shared medium. As a result, various scheduling algorithms, called *lazy scheduling*, have been proposed to transmit packets slowly within given constraints, e.g., a latency bound in [59, 44, 77].

CHAPTER III

A FRAMEWORK AND KNOWLEDGE-BASED MECHANISMS FOR A SINGLE RADIO ARCHITECTURE

3.1 Introduction

In this chapter, we develop a power management framework for a single radio architecture in which a node has one radio for contact discovery as well as communication in a DTN. DTNs are frequently partitioned, providing a chance for nodes to save energy when they are isolated from the rest of the network. However, the major issue in designing power management for DTNs is that a node needs to determine when it can communicate with other nodes, so that it can sleep for the rest of time. The time interval when two nodes can communicate with each other is called a *contact* [30]. When networks are partitioned most of the time, it is not trivial to discover contacts while also saving energy. In addition to this, a node may have overlapping contacts with multiple nodes that start and end at different times. Thus, a power management mechanism must take into account the aggregated state information of multiple contacts.

In our framework, a node is allowed to save energy while missing few contacts. Further, the framework supports an explicit trade-off between energy savings and connectivity, so that network operators can choose, for example, to conserve energy at the cost of reduced message delivery performance. In the basic framework, a node transits between three modes: a *dormant mode* (when no contact is expected), a *search mode* (when there is some expectation of a contact), and a *contact mode* (once a contact is established). The transitions between the modes are controlled by the available knowledge about network connectivity over time, in a fashion similar to the approach used to investigate DTN routing algorithms [30]. Specifically, we develop mechanisms for two extremes of knowledge: complete knowledge of contacts and no knowledge of contacts. These results establish the basic ideas and provide a context for the general and potentially more realistic case of partial knowledge. Within the category of partial knowledge, we develop mechanisms that use two forms of summary information about contacts (mean time between contacts and contact duration, and mean+variance time between contacts and contact duration). In addition to the framework, we devise a mechanism to enhance the performance of our power management by taking local traffic information into account.

We evaluate our mechanisms using ns-2 simulations with two node movement models and various scenarios for available knowledge. We find that our power management mechanisms reduce energy consumption from 60% to 93% compared with the case without power management. These energy savings come at the cost of some performance degradation when there is incomplete knowledge of contacts, though a tuning parameter for the partial knowledge model allows a rich set of trade-offs between energy consumption and performance. In addition, our investigation on various mobility scenarios illustrate that the details of the power-performance trade-off are sensitive to the scenario, and in particular to the ability of the partial knowledge to allow accurate prediction of future contacts. Finally, our traffic-aware enhancement saves additional energy on top of the energy savings through the main power management mechanisms.

The remainder of the chapter is structured as follows. Section 3.2 describes our system model. In Section 3.3, we present the overview of our power management framework as well as the detailed power management mechanisms according to the amount of available knowledge. Section 3.4 explains how to enhance the performance of power management using local traffic information. Section 3.5 presents our simulation results. We summarize the chapter in Section 3.6.

3.2 *System Model*

3.2.1 Network Topology

We consider a DTN consisting of mobile nodes as well as stationary nodes in a deployment area. The DTN topology changes dynamically because of mobile nodes or other factors. These dynamics can be represented as time varying states among nodes that represent whether any pair of nodes can communicate or not at time t. The time intervals when two nodes can communicate with each other are defined as *contacts* [30]. Thus, each pair of nodes alternates between a contact and waiting time for the next contact. Figure 2 illustrates an example scenario. Nodes move along their paths. Initially, the nodes are too far away from each other to communicate. From time t_0 to time t_1, the nodes are close enough to communicate. This time period is a contact between them. After t_1, the nodes are again too far away to communicate and then become close again at time t_2. This time period from t_1 to t_2 is a waiting time between the termination of one contact and the start of the next. The nodes then have a contact and waiting time alternatively.

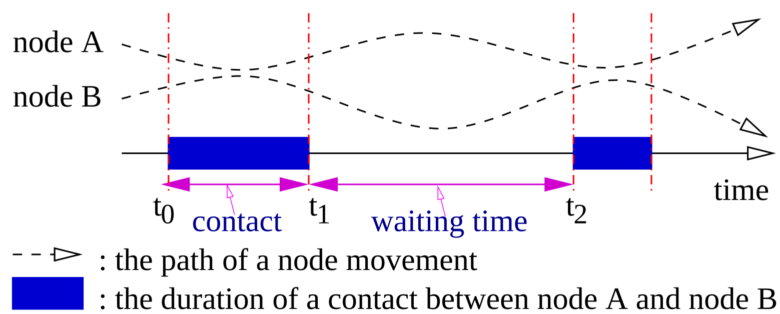


Figure 2: A node state that alternates between a contact and waiting for the next contact based on the distance from another node

3.2.2 Knowledge Model

We are interested in designing power management mechanisms that are applicable in a wide range of DTN settings. From the standpoint of mechanism design, the crucial information is the knowledge of future contacts. Because different networks will be capable of providing different levels of knowledge, we follow the approach of [30] and model knowledge of contacts using oracles. Specifically, we consider the following oracles:

Mean Oracle: This oracle can answer questions about the mean of contact duration and waiting time between contacts for every pair of nodes.

Variance Oracle: This oracle can answer questions about the variance of contact duration and waiting time between contacts for every pair of nodes.

Contact Oracle: This oracle can answer questions about the precise time when a contact starts and ends for every pair of nodes.

The realization of these oracles is not the focus of this chapter. However, we describe example scenarios to show the feasibility of their construction. For example, the mean and variance oracles can be built based on the history that nodes experience over time. This approach of the oracle construction is useful when nodes tend to have consistent mobility patterns. Also, the contact oracle can be obtained through calculation if the node movement is predetermined. This approach is feasible when mobile nodes are robots or vehicles that have predefined paths to move along ([78, 79]). In cases where these oracles can only be approximated, our results provide performance bounds under defined ideal conditions.

These oracles are used by power management mechanisms. Depending on the amount of knowledge they need, the mechanisms fall into the following classes:

Zero Knowledge: Mechanisms in this category do not use any oracle. The power management mechanisms in this category are simple to implement, and they provide the baseline for evaluating the performance in the presence of other information.

Partial knowledge: Mechanisms in this category use only the mean or both the mean and variance oracles. That is, only statistics of contact duration and waiting time are used. This partial knowledge improves the performance of mechanisms with zero knowledge.

Complete Knowledge: Mechanisms in this category use the contact oracle. Thus, the precise time of contacts is used rather than their statistics. This knowledge provides the upper bound on performance.

3.2.3 Energy Consumption

In this chapter, we consider only communication energy consumption and do not account for energy consumption of other sources such as computation or mobility. The energy consumption of a wireless interface depends on its activities: i.e., *transmitting*, *receiving*, *idling* (when listening to the wireless medium without transmitting or receiving), *dozing* (when the wireless interface is disabled, but powered), or *off*. Table 1 shows the power usage of a sample radio, DLink DCF-660W Wireless CompactFlash 802.11b card [15]. When dozing, a node consumes an order of magnitude less energy than when idling, while an

Table 1: Power usage of a DLink DCF-660W Wireless CompactFlash 802.11b card (unit:watt)

Activity	Transmit	Receive	Idle	Doze	Off
Power (W)	1.4349	1.3832	1.0649	0.2005	0

Table 2: Transition overhead of a DLink DCF-660W Wireless CompactFlash 802.11b card (unit:second, joule)

Action	Disable	Enable	Turn-off	Turn-on
Latency(s)	0.0078	0.0202	2.4001	2.5235
Energy (J)	0.0104	0.0221	2.3539	2.4738

idling node consumes energy at the same order of magnitude as a receiving or transmitting node. When a radio is not in use, energy can be saved by turning off or disabling the radio (i.e., placing a radio to doze). If turned off, the radio does not consume any power, yet the overhead to turn it on is significant. If disabled, the radio has lower overhead to enable, but it still consumes power. Table 2 shows the measured overhead of a DLink DCF-660W Wireless CompactFlash 802.11b card in terms of latency and energy consumption [15]. Both latency and energy consumption to turn off/on a DLink 802.11b card is two orders of magnitude higher than those to disable/enable the card.

3.3 Power Management Framework

In our framework, a node is in one of three power management modes: *dormant*, *search*, and *contact modes*. In the dormant mode, a node sleeps because it does not expect to have any contact. In the search mode, a node wakes up frequently to discover a contact. Finally, in the contact mode, a node stays awake to communicate with other nodes that it is in contact with.

The energy consumption in each mode decreases in the order of the contact, search, and dormant modes. As a result, the policy that determines when to transit among the power management modes determines the energy consumption of nodes. Thus, we utilize available knowledge to save energy by carefully changing the modes. In addition, when a radio sleeps, a node chooses to turn the radio completely off, disable it, or even leave it awake depending on the length of the expected time to sleep and the transition overhead. For example, if the expected time to sleep is greater than 23.94 seconds, the 802.11 card in table 1 should be completely off, instead of being disabled. The detailed algorithm can be found in Section 4.3.4. Moreover, a node may have overlapping contacts with multiple nodes that start and end at different times. Therefore, we aggregate the information about multiple contacts to decide when to transit from one mode to another.

In the rest of this chapter, we describe how to aggregate the information of multiple contacts and when to transit from one mode to another. Our objective is to save energy while discovering enough contacts to provide adequate data delay and throughput performance.

3.3.1 Power Management with Complete Knowledge

In this section, we describe our power management mechanism when complete knowledge about contacts is provided. Saving energy while discovering contacts is easily achievable with complete knowledge. In the case that a node has contacts with only one other node, the node wakes up at the beginning of a contact and sleeps at the end of the contact. In the case that a node has contacts with multiple nodes, the node wakes up at the earliest time when a contact starts and goes to sleep at the latest time when all the overlapping contacts end.

Figure 3 shows the transitions among power management modes after aggregating the state information of multiple contacts. Initially, a node is in the dormant mode. When a contact starts, the node enters the contact mode. When another contact starts, the node stays in the contact mode without any change. When a contact terminates, the node checks whether it has any other contacts. If so, the node stays in the contact mode. Otherwise, it enters the dormant mode.

3.3.2 Power Management with Zero Knowledge

In this section, we describe our power management mechanism when no knowledge is provided. In this class, a node has no idea when a contact starts and ends, so it needs to

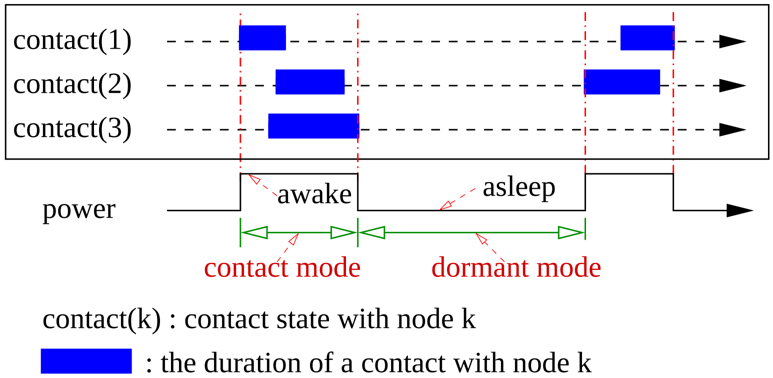


Figure 3: Transition among power management modes after aggregating multiple contact states in the complete knowledge class

discover contacts. To discover contacts, we use radio signals. That is, nodes frequently broadcast messages called *beacons*. If a node receives a beacon from another node that it is not in contact with yet, it considers that a new contact starts. The node keeps listening to beacons from the same node. If it fails to receive a certain number of beacons consecutively from the corresponding node, the node assumes that the contact is terminated.

To save energy while discovering contacts, we assume that nodes have synchronized clocks and can all use a common time period to control when to wake up. Specifically, nodes have three time periods: *beacon window*, *post-window space*, and *wake-up period*. A beacon window is a time period when a node randomly chooses a waiting time to send its own beacon to reduce contention. A post-window space is a small time period after a beacon window, in which a node listens to beacons beyond the beacon window. Finally, a wake-up period is the time between the beginning of two consecutive beacon windows.

In the case that a node has contacts with only one node, these time periods are used as shown in Figure 4. Initially, a node is in the search mode since it is not in contact with node *k*. In the beginning of a beacon window, it chooses a random time within the beacon window and broadcasts a beacon. If it does not receives a beacon, the node sleeps at the end of the post-window space and wakes up at the beginning of the next beacon window. Then, it repeats the procedure. If the node receives a beacon, it enters the contact mode. Finally, if the node does not receive a certain number of beacons, it considers that the contact is terminated and enters the search mode.

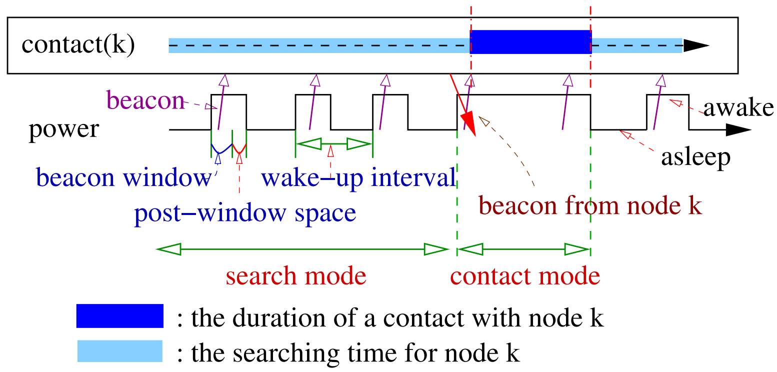


Figure 4: Transition among power management modes when a node has contacts with only one node, node *k* in the zero knowledge class

In the case that a node has contacts with multiple nodes, the node should consider multiple contact states to transit among the power management modes. Figure 5 shows an example scenario. When a node has no contact, it alternates between sleeping and waking up in the search mode. When it receives a beacon, it enters the contact mode. When it receives a beacon from another node, the node stays in the contact mode without any change. When it detects the termination of a contact, the node checks whether it has other contacts. If so, it stays in the contact mode. Otherwise, it enters the search mode.

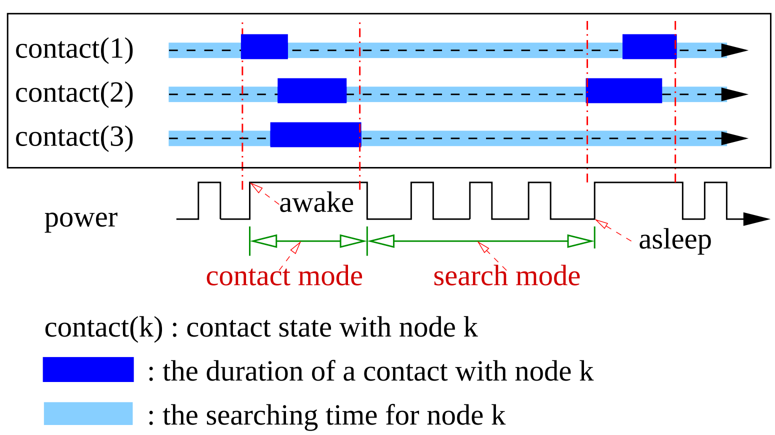


Figure 5: Transition among power management modes after aggregating multiple contact states in the zero knowledge class

To aggregate the state information of multiple contacts, a node manages a separate contact state per neighbor and one global parameter called `ContactCounter`. The `ContactCounter` represents the number of contacts that a node has at any point of time. To describe an individual contact state, we use two contact states based on node activities: *search state* when listening to beacons and *communication state* when having an established contact with the specific neighbor. Each contact state is managed based on beacon reception and timer expiration events, while transitions between power management modes occur based on `ContactCounter` change events.

Figure 6 describes the detailed procedure. A node has N contact states and one power management state when it has N neighbors that it can potentially have contacts with. Each contact state is managed individually as follows. Initially, a node enters the search state for the contact with node k. Then, it blocks and waits until receiving a beacon from node k. When it receives a beacon, it increments the `ContactCounter` and switches to the communication state. When it switches to the communication state, the node starts a timer to expire after a timeout value of a contact, τ_c. It then blocks and waits until either receiving a beacon from node k or the timeout of the timer. If the node receives beacons from node k, it continues the loop and resets the timer for τ_c. Otherwise, at the timeout, the node decrements the `ContactCounter` and leaves the loop in the communication state. As a result, the node continues on the main loop and enters the search state again.

While multiple contact states are managed individually, the transition among the power management modes is triggered by the changes of the `ContactCounter`. If the `ContactCounter` increases from 0 to 1, the node enters the contact mode because its first contact starts. If the `ContactCounter` decreases from 1 to 0, the node enters the search mode because it has no more contacts. For all the other changes of the `ContactCounter`, the node stays in the contact mode because it has one or more contacts.

3.3.3 Power Management with Partial Knowledge

In this section, we describe our power management mechanism when partial knowledge is provided. In the partial knowledge class, a node knows statistics about contact durations and waiting times between contacts with other nodes. With this information, a node can determine when to expect a contact. Depending on network characteristics, a node may

```
ManageContactState(k)
WHILE(true)
  state[k] ← SEARCH
  BlockedWaitUntilRecvBeacon(k)
  IncrementContactCounter()
  state[k] ← COMMUNICATION
  WHILE(true)
    reset a timer T[k] for τc
    event ← TimedWaitUntilRecvBeacon(k,T[k])
    IF event == RECV_BEACON_FROM_NODE_K
      CONTINUE
    ELSE IF event == TIMEOUT
      DecrementContactCounter()
      BREAK

IncrementContactCounter()
IF ContactCounter increases from 0 to 1
  EnterContactMode()
DecrementContactCounter()
IF ContactCounter decreases from 1 to 0
  EnterSearchMode()
```

Figure 6: Pseudo-code for a node to manage the contact state with neighbor k and to transit between power management modes in the zero knowledge class

expect to wait for a long time before a contact. In addition, it may have a short period of time when it expects to have a contact with high probability. Thus, if a node sleeps when the probability to have a contact is low and wakes up to search when the probability to have a contact is high, the node may save a significant amount of energy while discovering most of the contacts. However, a node may miss a few contacts because it does not search all the time. If it misses a contact, it stores messages until it can forward them in the next contact, which increases the delivery delay. Thus, deciding when to search trades between energy and delivery performance. Here, the time interval between when a node starts to search and when the node ends searching is called *searching interval*. In the rest of the section, we illustrate our power management mechanisms and then describe algorithms to determine when to search.

3.3.3.1 Mechanism Overview

In the partial knowledge case, a node uses the same basic mechanisms for contact discovery as in the zero knowledge case. That is, a node alternates between sleeping and waking up in the search mode until a contact is discovered; For searching, beacons are used to discover neighbors. In addition, a node may fail to discover a contact during a searching interval. Thus, a node has a series of searching intervals until it discovers a contact.

In the case that a node has contacts with only one node, the node transits among the power management modes based on its searching intervals. Figure 7 shows an example scenario. Initially, a node is in contact with node k and it terminates the contact at time t_0. At t_0, it enters the dormant mode and estimates a searching interval S_1. At the beginning of S_1, the node enters the search mode and periodically wakes up to discover a contact. If it does not discover a contact, it goes to sleep in the dormant mode at the end of S_1. Now, the node estimates the next time interval to search. Since its trial during S_1 has failed, it may select a longer period of time S_2 than S_1 to search for a contact. If it fails again during S_2, it may select an even longer period of time S_3 than S_2 to search for a contact. If it discovers a contact, it enters the contact mode and repeats the procedure at the end of the contact.

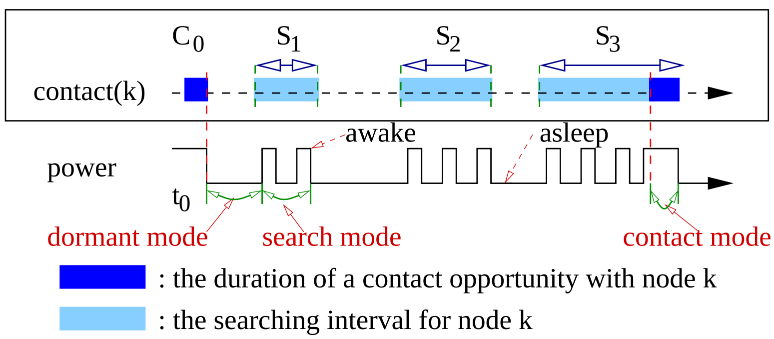


Figure 7: Transition among power management modes when a node has contacts with only one node, node k, in the partial knowledge class

In the case that a node has contacts with multiple nodes, the node should consider multiple contact states to transit among the power management modes. Figure 8 shows an example scenario. Initially, a node estimates searching intervals for every potential neighbor node and enters the dormant mode. At the earliest time when a searching interval starts, the node enters the search mode. When another searching interval starts, it stays in the search mode. In the search mode, if the node receives a beacon, it enters the contact mode. If it receives beacons from other nodes, it stays in the contact mode. When a contact terminates, the node checks whether it has other contacts. If so, it stays in the contact mode. Otherwise, it checks whether it has any active searching intervals. If so, it enters the search mode. Otherwise, it enters the dormant mode.

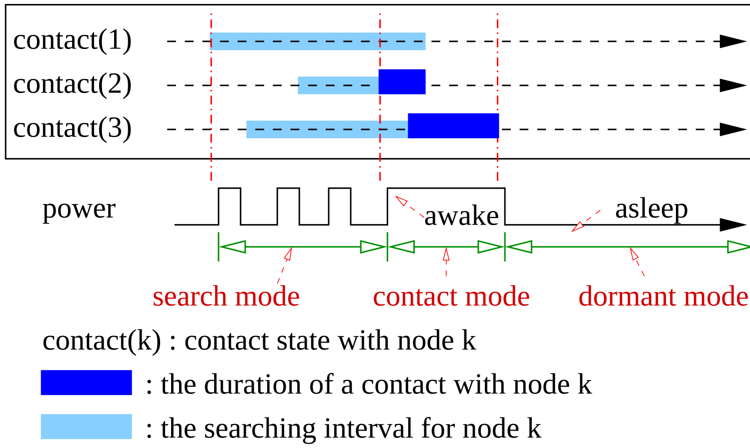


Figure 8: Transition among power management modes after aggregating multiple contact states in the partial knowledge class

To aggregate the state information of multiple contacts, a node manages a separate contact state per neighbor and two global parameters: `ContactCounter` and `SearchCounter`, where the `SearchCounter` represents the number of active searching intervals that a node has at one point of time. To describe an individual contact state, we use three contact states based on node activities: *search state*, *communication state*, and *hibernation state* when not actively listening to beacons. In addition, the node has two additional time periods: τ_h and τ_s. The period τ_h is the time in the hibernation state until a searching interval starts, and the period τ_s is the duration of a searching interval. Each contact state is managed based on beacon reception and timer expiration events, while transitions between power management modes occur based on `ContactCounter` and `SearchCounter` change events.

Figures 9 and 10 describe the detailed procedure. A node has N contact states and one power management state when it has N neighbors it can potentially have contacts with. Each contact state is managed individually as follows. Initially, a node starts a loop in the hibernation state for the contact with node k. In the hibernation state, the node estimates τ_h and τ_s. After setting a timer to expire after τ_h, it blocks and waits until either receiving a beacon from node k or the timeout of the timer. If the node receives a beacon, it increments the `ContactCounter` and switches to the communication state. This can occur if the node is awake to receive beacons for other contacts. Otherwise, at the timeout, the node increments the `SearchCounter` and switches to the search state. In the search state, the node sets a timer to expire after τ_s and waits until either receiving a beacon from node k or the timeout of the timer. If the node receives a beacon, it increments the `ContactCounter` and switches to the communication state. Otherwise, at the timeout, the node decrements the `SearchCounter` and continues on the main loop to enter the hibernation state. In the communication state, the node starts a loop by setting a timer to expire after a timeout value of a contact, τ_c. Then, it waits until either receiving a beacon from node k or the timeout of the timer. If the node receives additional beacons from node k, it continues the loop and refreshes the corresponding timer. Otherwise, at the timeout, the node decrements the `ContactCounter` and leaves the loop of the communication state. As a result, the node continues on the main loop and enters the hibernation state.

ManageContactState(k)

```
WHILE(true)
  state[k] ← HIBERNATION
  estimate τ_h and τ_s
  reset a timer T[k] for τ_h
  event ← TimedWaitUntilRecvBeacon(k,T[k])
  IF event == RECV_BEACON_FROM_NODE_K
    IncrementSearchCounter()
    GOTO COMMUNICATION
  ELSE IF event == TIMEOUT
    IncrementSearchCounter()
  SEARCH:
    state[k] ← SEARCH
    reset T[k] for τ_s
    event ← TimedWaitUntilRecvBeacon(k,T[k])
    IF event == RECV_BEACON_FROM_NODE_K
      IncrementContactCounter()
      GOTO COMMUNICATION
    ELSE IF event == TIMEOUT
      DecrementSearchCounter()
      CONTINUE
  COMMUNICATION:
    state[k] ← COMMUNICATION
    WHILE(true)
      reset T[k] for τ_c
      event ← TimedWaitUntilRecvBeacon(k,T[k])
      IF event == RECV_BEACON_FROM_NODE_K
        CONTINUE
      IF event == TIMEOUT
        DecrementContactCounter()
        BREAK
```

Figure 9: Pseudo-code for a node to manage the contact state with neighbor k in the partial knowledge class

```
IncrementSearchCounter()
1. IF SearchCounter increases from 0 to 1
2.   EnterSearchMode()
DecrementSearchCounter()
1. IF SearchCounter decreases from 1 to 0
2.   EnterDormantMode()
IncrementContactCounter()
1. IF ContactCounter increases from 0 to 1
2.   EnterContactMode()
DecrementContactCounter()
1. ContactCounter--, SearchCounter--
2. IF ContactCounter decreases from 1 to 0
3.   IF SearchCounter > 0
4.     EnterSearchMode()
5.   ELSE
6.     EnterDormantMode()
```

Figure 10: Pseudo-code for a node to transit between power management modes in the partial knowledge class

While multiple contact states are managed individually, the transition among the power management modes is triggered by the changes of the `ContactCounter` and the `SearchCounter`. If the `SearchCounter` increases from 0 to 1, the node enters the search mode because it needs to start searching for a contact. If the `SearchCounter` decreases from 1 to 0, the node enters the dormant mode because it does not need to search for a contact any more. If the `ContactCounter` increases from 0 to 1, the node enters the contact mode because its first contact starts. Finally, when a node decrements the `ContactCounter`, it also decrements the `SearchCounter`. If the `ContactCounter` decreases from 1 to 0, the node checks whether the `SearchCounter` is greater than zero. If so, it enters the search mode because it needs to search for another contact. Otherwise, it enters the dormant mode because it has no contact and no active search interval.

3.3.3.2 Searching Interval Estimation Overview

The critical issue in the operation of this mechanism is the scheduling of searches. In DTNs, a pair of nodes has a series of contacts. Since we have statistics of contact duration and waiting time between contacts, we can determine when to expect a contact. Thus, we estimate searching intervals based on the expected time of each contact. Figure 11 shows an example scenario. Initially, a node has a contact with Node k and terminates the contact at time t_0. Then, the node estimates a searching interval S_1 around the expected time of the first contact since t_0. If it fails to discover a contact during S_1, it selects the next searching interval S_2 around the expected time of the second contact since t_0. Similarly, if the node fails to discover a contact during $(n-1)$ searching intervals, it selects a searching interval S_n around the expected time of the nth contact since t_0.

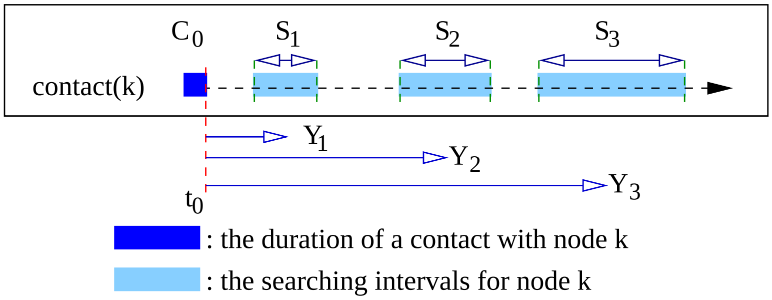


Figure 11: The estimation of searching intervals for a node

To describe the detailed procedure, we denote X_i as i.i.d. random variables of waiting times between contacts and C_i as i.i.d. random variables of contact durations for two nodes. Define Y_n as the expected time when the nth contact opportunity begins since the last contact termination time. Then, Y_n consists of n waiting times and $(n-1)$ contact durations. That is,

$$Y_n = \sum_{k=1}^{n} X_k + \sum_{k=1}^{n-1} C_k. \tag{1}$$

The last contact termination time is denoted as t_0, which is assumed to be zero without loss of generality. A searching interval since the last contact termination time is denoted as S_n, which is

$$S_n = [t_{n,1}, t_{n,2}], \tag{2}$$

where $0 \leq t_{n,1} < t_{n,2}$ for all $n \geq 1$.

In addition to the available statistical information, we include a tuning parameter p that allows the network operator to specify a desired ratio to discover contacts over the total contacts. Thus, we define a *contact discovery ratio* p as follows:

$$p = \frac{the\ number\ of\ discovered\ contacts}{the\ total\ number\ of\ contacts}. \tag{3}$$

When p is small, energy will be saved at the expense of missed contacts. When p is large, energy will be expended to discover more contacts. We then state the problem as follows: *Find minimal searching intervals S_n that discover contacts at least at the ratio of p.*

3.3.3.3 Estimation with the Mean Oracle

This oracle provides only the mean values of contact duration and waiting time between contacts for every two nodes. According to the central limit theorem [61], the distribution of the summation of i.i.d. random variables becomes indistinguishable from the normal distribution for large number of variables. Therefore, an interval around the mean of Y_n will provide high probability to discover a contact for large n. In addition, for independent random variables, the mean of their summation is the summation of their means. That is, the mean μ_n of Y_n is as follows:

$$\mu_n = n\mu_X + (n-1)\mu_C, \tag{4}$$

where the mean of X_k is μ_X and that of C_k is μ_C for all k.

With this intuition, we propose a heuristic method to select S_n as follows:

$$S_n = [\mu_n - \alpha\mu_n,\ \mu_n + \alpha\mu_n], \tag{5}$$

where α is a constant. Here the contact discovery ratio constraint is satisfied as follows. Since μ_n increases as n increases, the length of S_n, $2\alpha\mu_n$, increases as n increases. Thus, for a large n, S_n overlaps with the next interval S_{n+1}. That is, where $S_n = [t_{n,1}, t_{n,2}]$, $t_{n+1,1} - t_{n,2} \leq 0$ for all $n \geq \frac{1}{2\alpha} + \frac{1}{2}$. Thus, S_n overlaps with the next interval S_{n+1} for all n, where $n \geq \hat{n} = \lceil \frac{1}{2\alpha} + \frac{1}{2} \rceil$. In other words, after the $\hat{n}$th searching interval, the nodes search all the time until a contact is discovered. Thus, the maximum number of searching intervals until a contact discovery is $\hat{n}$, so that the contact discovery ratio can be approximated as $\frac{2\alpha}{\alpha+1}$. Therefore, to discover contacts at least at the desired ratio of p, α should be as follows:

$$\alpha = \frac{p}{2-p}, \tag{6}$$

where $0 \leq p \leq 1$. As a result, $0 \leq \alpha \leq 1$. Therefore, we could reduce the length of searching intervals while discovering contacts at the ratio of p, approximatively.

3.3.3.4 *Estimation with the Mean and Variance Oracles*

This oracle provides both the mean and variance values of contact duration and waiting time between contacts for every two nodes.

With this information, we can apply Chebyshev's inequality [61] to obtain an interval that achieves the contact discovery ratio p. In addition, for independent random variables, the variance of their summation is the summation of their variances. That is, the variance σ_n^2 of Y_n is as follows:

$$\sigma_n^2 = n\sigma_X^2 + (n-1)\sigma_C^2, \tag{7}$$

where the standard deviation of X_k is σ_X and that of C_k is σ_C for all k. Then, Chebyshev's inequality for Y_n can be restated as follows:

$$P\left[|Y_n - \mu_n| < \beta\sigma_n\right] \geq 1 - \frac{1}{\beta^2}, \tag{8}$$

for all $\beta > 0$.

With these facts, we propose a heuristic method to select S_n as follows:

$$S_n = [\mu_n - \beta\sigma_n,\ \mu_n + \beta\sigma_n], \tag{9}$$

where μ_n and σ_n are the mean and the standard deviation of Y_n, and β is a positive real number. Observe that S_n gets longer as n increases. This is because the uncertainty was quantified by increasing variance reflected in Equation 7. Here, the contact discovery ratio constraint is satisfied as follows. Substitute the right side of Equation 8 with p. Then, it implies that the probability to discover the nth contact at the nth searching interval is greater than or equal to p. Here the expected number of trials until discovering the nth contact at the nth trial is

$$\sum_{k=1}^{\infty} kp(1-p)^{k-1} = \frac{1}{p}. \tag{10}$$

Since we do not limit ourselves to discover only the specific contact at each trial, our expected number of trials until discovering a contact is less than $\frac{1}{p}$. Therefore, the contact discovery ratio is at least p. Also, we obtain

$$\beta = \frac{1}{\sqrt{1-p}}, \tag{11}$$

where $0 \leq p < 1$. Therefore, we could again reduce the length of searching intervals while the contact discovery ratio is at least p.

3.3.4 Selection of Desired Contact Discovery Ratio

The desired contact discovery ratio p is an important tuning parameter between energy and delivery performance. When p is small, nodes can save energy at the cost of missing contacts, which results in poor delivery performance. This parameter p can be selected based on various network management criteria such as traffic load, remaining energy, target network lifetime, routing policies, etc. For example, if the traffic generation rate is greater than the delivery rate, the node will eventually drop messages due to its limited buffer space. Thus, nodes may need to discover enough contacts to handle any expected traffic load at the cost of energy. In this section, we describe a mechanism to determine p for each pair of nodes to discover enough contacts to deliver the expected traffic load while minimizing energy consumption, where traffic load among nodes can be estimated in advance.

To formulate the problem, we assume that the statistical information about contacts and traffic load is already available in a DTN: i.e., nodes observe and exchange the history of traffic load to make efficient routing decision [25, 18]. We define the *contact arrival rate* as the number of contacts between two nodes over a unit time and *expected bandwidth* as the maximum amount of data that can be delivered by the discovered contacts between two nodes over a unit time. We also assume that contacts arrive according to a Poisson process. In Section 5.2.1, we describe an analytical method to estimate the expected discovered contact time $C(w)$ for a given wake-up period w. Using that result, for a pair of nodes, the expected bandwidth is $\lambda \cdot \theta \cdot C(w) \cdot p$ for a given p, where λ is the contact arrival rate and θ is the bandwidth of the radio. Then, we need to find p that provide bandwidth greater than or equal to the traffic load between the nodes. That is, p should satisfy the following equation:

$$\lambda \cdot \theta \cdot C(w) \cdot p \geq \tau, \tag{12}$$

where τ is the traffic load between the two nodes over a unit time. Thus, the parameter p

that minimizes the energy consumption while discovering enough contact time to handle the expected traffic load is

$$\frac{\tau}{\lambda \cdot \theta \cdot C(w)}. \tag{13}$$

We leave further investigation on the select of p with other criteria as future work.

3.4 Traffic-Aware Enhancement

Our power management framework has focused on saving energy expended in searching for contacts. Although the energy consumption while searching for contacts is the major issue in DTNs, the framework can be enhanced by taking traffic load between nodes in contact into account. For example, if two nodes have no message to exchange when they discover their contact, they can save more energy by putting their radios to sleep for the rest of the contact. In fact, power management with complete or zero knowledge leads for a node to discover most of the contacts regardless of traffic load. Power management with partial knowledge also leads for a node to discover more than a desired ratio even if the desired ratio is set based on the expected traffic load. Therefore, additional energy can be saved by employing traffic awareness in the contact mode. For this purpose, we use link management between nodes in contact to save energy based on their local traffic load. That is, a node establishes a link to its party only when they have messages to exchange. When they finish exchanging messages, the node terminates the link and its power is managed based on the link status in the contact mode. Except for the power management in the contact mode, the node uses the same mechanisms such as contact discovery, contact management, and power management in the other modes. The amount of energy saved while searching or having contacts will be investigated in Section 3.5.4. In the remaining section, we describe how to enhance power management in the contact mode in a partial knowledge case. It can be extended to the other knowledge cases similarly.

In the case that a node contacts only one node, the node wakes up before the initiation of the link and sleeps at the termination of the link in the contact mode. Figure 12 shows an example scenario in a partial knowledge case. Initially, a node is searching for a contact with node k. When it discovers a contact by receiving a beacon, it sends an advertisement signal to node k to tell how many messages it has for node k. It also receives an advertisement or a response from node k containing the number of messages to receive from node k. If there is no message to exchange, they terminate the link immediately. If there is any, they establish a link and start transmission. When they finish exchanging messages, they send signals to terminate the link. Also, a timer is set to expire after a certain timeout since the last message exchange and to terminate the link. When the link is terminated, the node puts its radio to sleep. While the node is sleeping, the node or node k may generate new messages to exchange. Also, the termination of their contact is detected by monitoring beacons while they are in contact. Thus, the node wakes up at the next wake-up interval to send a beacon and repeats the link establishment procedure.

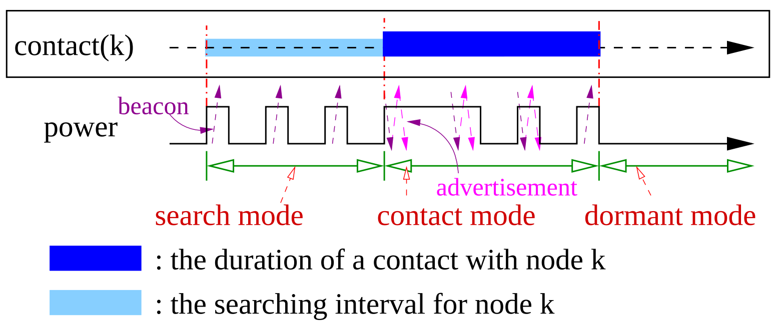


Figure 12: Transition among power states when a node has contacts with only one node, node k, in the enhanced power management of the partial knowledge class

In the case that a node has contacts with multiple nodes, the node should consider multiple link and contact states to manage its power. Figure 13 shows an example scenario. Initially, a node is in the dormant mode. At the earliest time when a searching interval starts, the node enters the search mode. When it discovers a contact, it enters the contact mode and establish a link. If it discovers a contact with another node, it establishes a link to that node. When both links are terminated, the node sleeps until the next wake-up interval. When it wakes up, it sends a beacon and advertisement signals to its neighbor nodes in contact. Since the node has no messages to send or to receive, it terminates its links immediately and sleeps until the next wake-up interval. The node repeats the procedure until all the contacts are terminated. This behaviors complies with its search for a contact with node 1. When both contacts are terminated, the node enters the search mode since it has one active searching interval. When the searching interval ends without discovering a contact, the node enters the dormant mode.

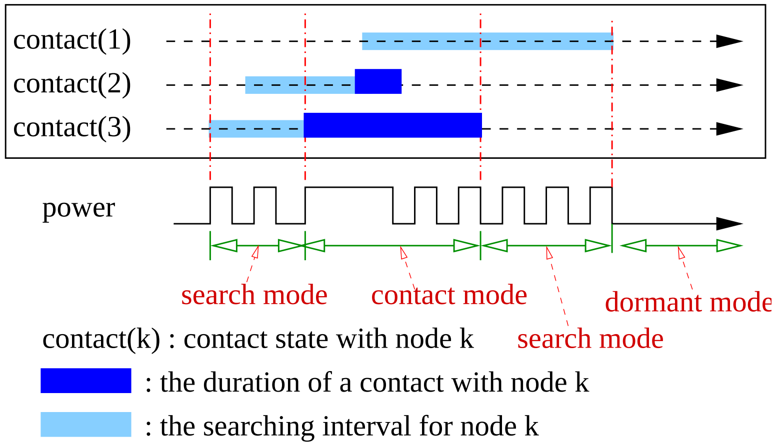


Figure 13: Transition among power states after aggregating multiple link and contact states in the enhanced power management of the partial knowledge class

To aggregate the link state information, a node manages one additional parameter: `linkCounter` which represents the number of active links. Also, the node has two additional time periods: τ_a and τ_l. The period τ_a is the timeout for a node to wait for a response from its party after sending its advertisement, and τ_l is the timeout for a link. Figure2 14 and 15 describe how these additional parameters are used to manage power in a partial knowledge case. To avoid duplicate explanation, we describe only the different parts in the state management. A node has N contact and link states and one power management state when it has N neighbors it can potentially have contacts with. Each state is managed individually as follows. Initially, a node is in the search state for node k. When it receives a beacon or an advertisement from node k, it switches to the communication state for the contact with node k. In the communication state, the node initializes its link state with node k and then starts a loop. In the loop, the node initiates the link and sets a contact timer to expire if it does not receive a certain number of beacons consequently. If it receives a beacon, it checks the link state. If the state is initiated, the node reschedules the contact timer without any other change. If the state is terminated, it initiates the link again. If the contact timer expires, the node terminates the corresponding link and switches to either search or hibernation state. On the other hand, the link state is managed separately.

```
ManageContactState(k)
WHILE(true)
  //same as in Figure 9
  SEARCH:
    IF RECV_BEACON or RECV_ADV_FROM_NODE_K
      GOTO COMMUNICATION
  //same as in Figure 9
  COMMUNICATION:
    state[k] ← COMMUNICATION
    link_state[k] ← TERMINATED
    WHILE(true)
      IF link_state[k] == TERMINATED
        InitiateLink(k)
      reset T[k] for τc
      event ← TimedWaitUntilRecvBeacon(k,T[k])
      IF event == RECV_BEACON_FROM_NODE_K
        CONTINUE
      ELSE IF event == TIMEOUT
        TerminateLink(k)
        DecrementContactCounter()
        BREAK
```

Figure 14: Pseudo-code for a node to manage the contact state with neighbor k in the enhanced power management of the partial knowledge class

When a link with a node k is initiated, the node increases its `linkCounter` and sends an advertisement to node k with the information about its traffic to node k. While it has not received an advertisement from node k yet, it sets a timer to expire after τ_a. At the timeout, the node resends the advertisement. When it receives an advertisement from node k or it sent the maximum number of advertisement, the node investigates how many messages they have for each other. If none, the node terminates the link immediately. Otherwise, the node sets a link timer for τ_l and starts sending or receiving messages. Whenever the node sends or receives a message, the link timer is reset. When there is no message to exchange, the link timer expires at its timeout and the node terminates the link. When a link is terminated, the node decreases its `linkCounter`. When the `linkCounter` decreases from 1 to 0, the node estimates the remaining time until the next wake-up interval and puts its radio to sleep for the amount of time.

Each state is managed based on beacon and advertisement reception and timer expiration events, while transitions between power states within the contact mode occur based on the `linkCounter`. If the `linkCounter` decreases from 1 to 0, the node sleeps until the

```
InitiateLink(k)
link_state[k] ← INITIATED, numRetry ← 0
IncrementLinkCounter(), sendAdv(k)
WHILE(NOT_RECEIVED_AN_ADV
      && numRetry++ < NUM_MAX_RETRY)
  reset L[k] for τ_a
  event ← TimedWaitUntilRecvAdv(k,L[k])
  IF event == TIMEOUT, sendAdv(k)
  ELSE, BREAK
IF trafficLoad[k] == 0
  TerminateLink(k)
ELSE
  WHILE (trafficLoad[k] > 0)
    reset L[k] for τ_l
    event ← TimedWaitUntilSendRecvMessage(k)
    IF event == SEND_OR_RECV_MSG_FROM_NODE_K
      updateTrafficLoad(k)
    ELSE IF event == TIMEOUT
      BREAK
  TerminateLink(k)
TerminateLink(k)
link_state[k] ← TERMINATED
DecrementLinkCounter()
DecrementLinkCounter()
IF LinkCounter decreases from 1 to 0
  sleepUntilNextWakeUpInterval
DecrementContactCounter()
//same as in Figure 10
IF ContactCounter decreases from 1 to 0
  calcelLinkTimers()
//same as in Figure 10
```

Figure 15: Pseudo-code for a node to manage the link state with neighbor k and to transit power states in the enhanced power management of the partial knowledge class

next wake-up interval in the contact mode. For all the other changes of the `linkCounter`, the node stays awake because it has at least one active link.

3.5 *Performance Evaluation*

In this section, we evaluate the power management mechanisms presented in this chapter using ns-2 simulations [2]. We investigate how the availability of knowledge about network topology and the network characteristics affect the performance of power management. We also evaluate how the local information about the traffic load can be used to enhance the performance of power management.

3.5.1 Simulation Methodology

We implement the power management mechanisms using knowledge oracles presented in this chapter. Also, we implement the case without power management to compare its energy consumption with that of our power management. We consider the following three metrics:

- *normalized energy consumption*: the total energy consumption in a network divided by that of the case without power management,
- *delivery delay*: an average delay per delivered message,
- *delivery rate*: the ratio of successfully delivered messages to (the total number of generated messages - the number of remaining messages in the network) by the end of a simulation,

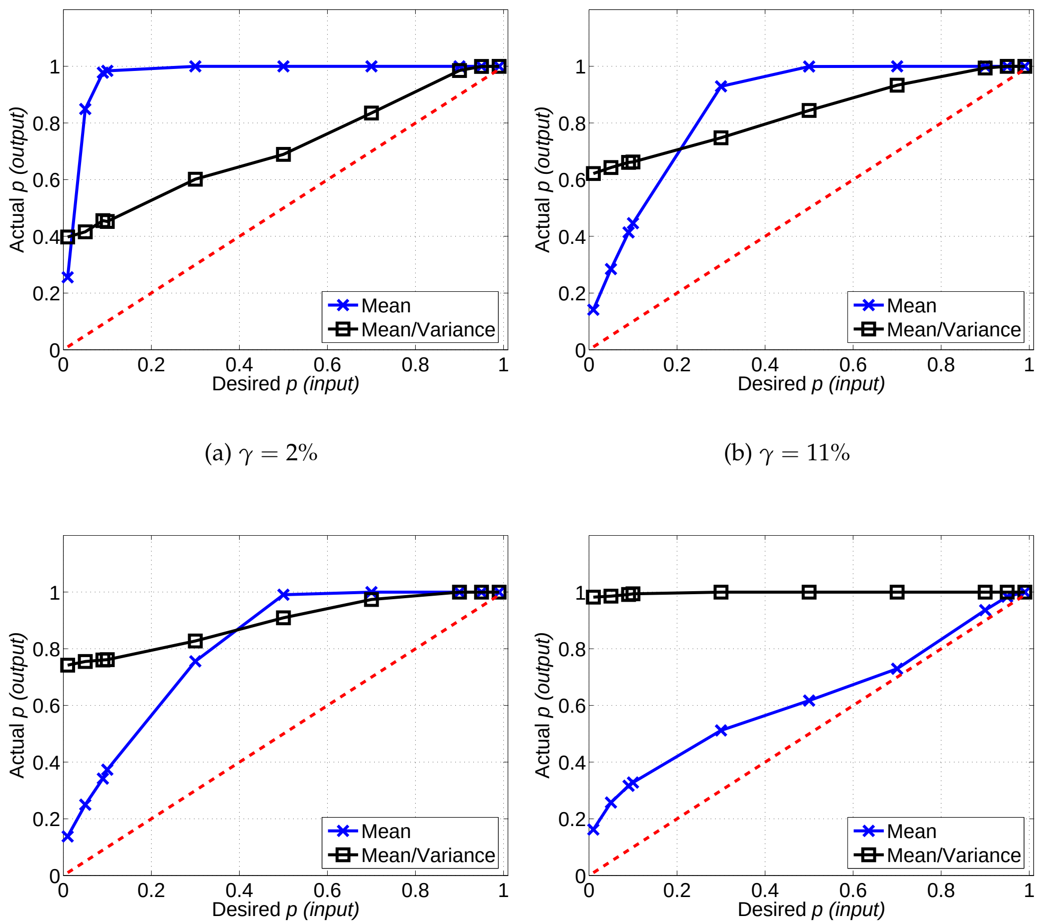
Here a message loss occurs when a message is not delivered within a given timeout, or when a message buffer overflows at a node in the middle of a routing path.

To determine the routing paths of messages, we use the modified Dijkstra's algorithms in [30] that find the shortest path to minimize the delivery delay of messages. Depending on the available knowledge of contacts, we use the first contact (FC) for the zero knowledge, the minimum expected delay (MED) for the partial knowledge, and the earliest delivery (ED) for the complete knowledge class. The resulting routing paths are also stored in the message headers to be used in source routing for MED and ED. Details can be found in [30].

In simulations, we use the following default parameters, unless specified otherwise. We use a 802.11 MAC layer. The radio range is 250m and the data rate is 2Mb/s. The power and transition overhead are set based on Table 1 and 2. In power management mechanisms, the wake-up period is 2 seconds, the beacon window is 30ms, and the post-window space is 270ms. For traffic load, each node generates 1KB messages according to a Poisson process with rate of one message per 100 seconds. The destination of each message is assigned differently depending on node movement scenarios as will be described shortly. Each message has 5 hours timeout, so that a message is discarded if it becomes older than the timeout value before reaching its destination. Also, each node has a limited buffer to store 5000 messages per neighbor node. To implement the knowledge oracles, we generate the movement of nodes for the whole simulation duration in advance and construct the oracles. Then, we use the same node movement to run simulations. Each simulation runs for 100 days in simulation time. Also, we warm up the simulations before measuring performance metrics. That is, for each pair of nodes that have contacts within the simulation time, the nodes starts searching a contact from the beginning with an infinite searching interval size. Once the nodes find a contact, they start their assigned power management, e.g., using the mean oracle. At the last time to discover the first contact between each pair of nodes, we reset energy and contact discovery related parameters, i.e., initial energy, the total number of contacts, and the number of discovered contacts.

To simulate node movement, we consider two node movement scenarios: Message Ferrying model (MF) and Random Way-Point model (RWP). These models have the following characteristics. The MF model provides a certain degree of regularity to the waiting time between contacts, while allowing randomness. Also, the degree of regularity of the MF model can be tuned from high to low. On the other hand, the RWP model provides a high degree of randomness to the waiting time between contacts. We model them as follows. In MF, a network consists of nine stationary nodes and one mobile node, called a *ferry*, in a 8km$\times$8km deployment area. To locate stationary nodes, we assign nine 200m$\times$200m zones sparsely in the deployment area and place one node at a random location of each zone. Then, the ferry visits the center of each zone in order and repeats the route. As a result, stationary nodes are too far away to communicate with each other and only the ferry provides network connectivity by visiting the neighborhood of stationary nodes from time to time. For traffic, all messages are destined to the ferry. In RWP, a network consists of 10 mobile nodes, which move in the Random Way-Point model [34], in a 5km$\times$5km deployment area. For traffic, 10 pairs of source and destination nodes are selected, and a source node sends messages toward its destination node. In both models, a mobile node selects a random speed between 9m/s and 10m/s and moves toward its destination.[1] When it reaches a destination, it pauses there for a *pause time* that is exponentially distributed with a given average: i.e., 200 seconds in MF and 30 seconds in RWP. When the pause time is up, the node moves toward the next destination.

3.5.2 Contact Discovery Ratio



(a) $\gamma = 2\%$

(b) $\gamma = 11\%$

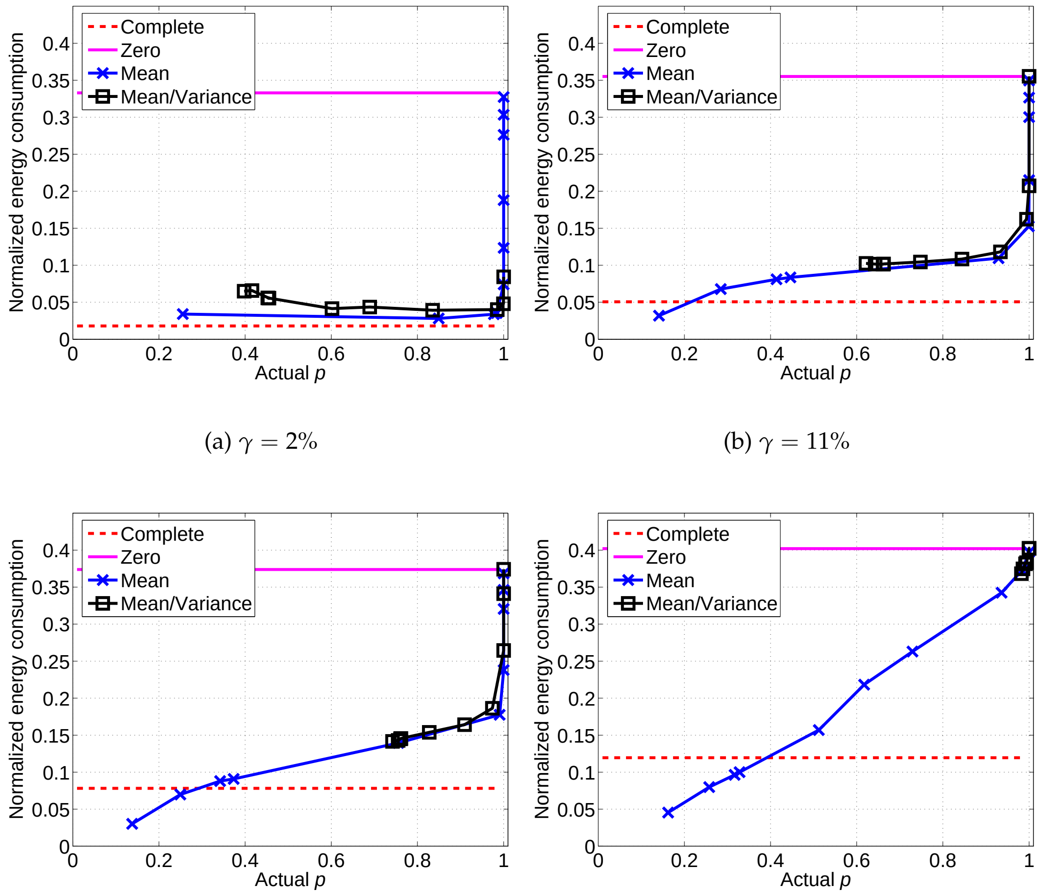
(c) $\gamma = 20\%$

(d) $\gamma = 87\%$

Figure 16: Actual contact discovery of power management mechanisms in four MF scenarios where γ represents the ratio of a standard deviation to the mean of contact waiting time in the corresponding scenario

In this section, we investigate the impact of our tuning parameter, the desired contact

[1]We use a non-zero minimum speed to adjust the stability problem of RWP as described in [76].



(a) $\gamma = 2\%$

(b) $\gamma = 11\%$

(c) $\gamma = 20\%$

(d) $\gamma = 87\%$

Figure 17: Energy consumption of power management mechanisms in four MF scenarios where γ represents the ratio of a standard deviation to the mean of contact waiting time in the corresponding scenario

discovery ratio p, to the contact discovery performance of power management mechanisms in four MF scenarios. The four scenarios are generated by varying the size of deployment areas and average pause times in the MF model: i.e., (1) 10km×10km with an average pause time of 30 seconds, (2) 8km×8km with an average pause time of 200 seconds, (3) 5km×5km with an average pause time of 300 seconds, and (4) 2km×2km with an average pause time of 700 seconds. Each scenario has one stationary node and one ferry. All scenarios have their mean contact waiting time within the range of 4664 ± 348 seconds, except for the last one, which has 2998 seconds. The ratio of a standard deviation to the corresponding mean of contact waiting time is denoted as γ. As γ decreases, nodes meet each other more regularly. As γ increases, nodes meet each other more randomly. The γ in the four scenarios is 2, 11, 20, and 89%, respectively. Thus, these scenarios can be used to explore the contact discovery performance of power management mechanisms in a spectrum of mobility characteristics from regular to random mobility.

Figure 16 shows the actual contact discovery ratio of power management in the four MF scenarios as the desired contact discovery ratio p varies from 0.01 to 0.99. Each line represents the performance of one partial knowledge case as indicated: partial knowledge with the mean oracle and partial knowledge with the mean and variance oracles. Also, a dashed line is placed for comparison, which represents when the actual p is equal to the input p. In all four scenarios, the actual contact discovery ratio is greater than the desired discovery ratio for both mean and mean/variance oracle cases. This is desirable because searching intervals are estimated to discover contacts at least at the input contact discovery ratio. Meanwhile, the sensitivity of contact discovery performance of each power management mechanism depends on mobility scenarios.

Figure 16(a) shows that the actual contact discovery ratio where γ is 2%. While the actual contact discovery ratio increases monotonically in both mean and mean/variance oracle cases, the actual contact discovery ratio of the mean/variance oracle case is closer to the dashed line than that of the mean oracle case because of their searching interval sizes. That is, if a searching interval is larger than a necessary size to discover contacts at the input contact discovery ratio, nodes will discover more contacts as a result of more searching. In the mean/variance oracle case, the size of a searching interval for a given p is determined by the standard deviations of contact waiting time and contact duration. Since the standard deviation of the contact waiting time is only 2% of the mean waiting time, the searching intervals are narrow enough to discover contacts almost at the desired discovery ratio without much more accidental discovery. In the mean oracle case, the sizes of searching intervals are proportional to the mean of contact waiting time and duration. As a result, the sizes of searching intervals are larger than that in the mean/variance oracle case. Thus, nodes in the mean oracle case discovers much more contacts than the desired contact discovery ratio even for as small p as 0.1.

Figure 16 shows the actual contact discovery ratio as γ increases as in 2, 11, 20, and 89%. As γ increases, the actual contact discovery of the mean/variance oracle case moves away from the dashed lines. On the other hand, the actual contact discovery ratio of the mean oracle case moves toward the dashed line. In the mean/variance oracle case, Chebyshev's inequality is used to estimate the searching intervals so that it guarantees to discover contacts at least at the input ratio regardless of contact waiting time distribution. As a result, as γ increases, the sizes of searching intervals for a given p increase and nodes discover more contacts than the input contact discovery ratio. In the mean oracle case, the sizes of searching intervals are proportional only to the mean of contact waiting time and contact duration. Thus, γ or the level of randomness in the node mobility does not affect the sizes of searching intervals. Meanwhile, as γ increases, nodes' meeting events become memoryless. Thus, their meeting events become distributed uniformly over time as in the Poisson process. Thus, in our searching interval estimation of the mean oracle case, the probability to discover a contact at each trial is the same as the desired contact discovery ratio. As a result, the actual discovery ratio is close to the dashed line when γ is high.

Figure 17 shows the normalized energy consumption of power management mechanisms in the same four MF scenarios as the desired contact discovery ratio p varies from 0.01 to 0.99. X-axis represents the actual contact discovery ratio, and y-axis represents the normalized energy consumption. Each line represents the performance of one of four knowledge cases as indicated: complete knowledge, zero knowledge, partial knowledge with the mean oracle, and partial knowledge with the mean and variance oracles.

Figure 17(a) shows that all mechanisms consume energy less than 35% of the case without power management. The energy consumption decreases in the order of the zero, partial, and complete knowledge cases. Thus, power management mechanisms with more knowledge save more energy in general. Also, the energy consumption of the partial knowledge cases is as low as that of the complete knowledge case when the actual discovery ratio is small. Especially, when the actual contact discovery ratio is less than 0.95, the energy consumption of both mean and mean/variance oracle cases is less than 93% of energy consumption of the case without power management and less than 75% of energy consumption of the zero knowledge case. Thus, when node mobility is regular, partial knowledge such as mean and variance of contact waiting time and contact duration is useful to narrow searching intervals and achieve energy savings equivalent to the complete knowledge case.

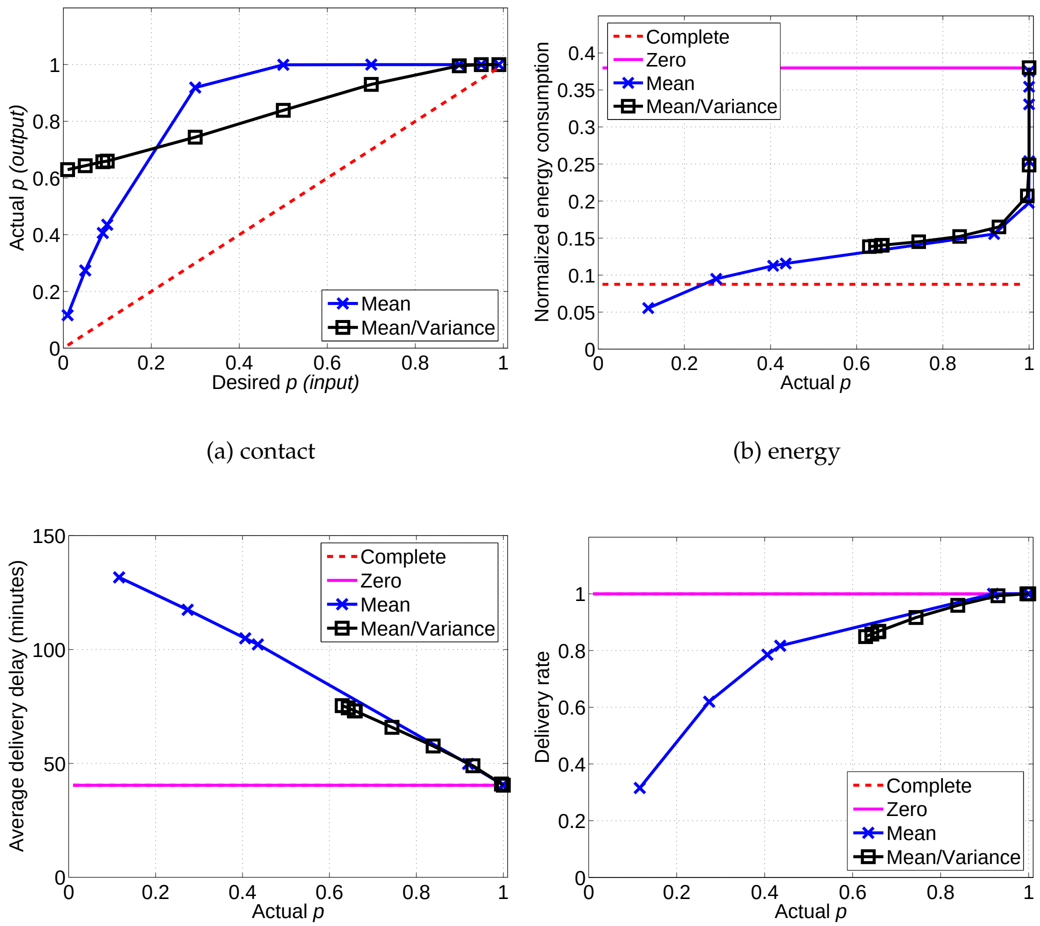
Figure 17(a) also shows that when the actual contact discovery ratio increases beyond 0.95, the energy consumption of the mean/variance oracle case increases only up to 10% of the energy consumption of the case without power management, while the energy consumption of the mean oracle case keeps increasing up to that of the zero knowledge case. In the mean/variance oracle case, Chebyshev's inequality determines the size of searching interval by the standard deviations of contact waiting time and contact duration. Since γ is only 2%, the estimated searching intervals are narrow for all input contact discovery ratio p. As a result, nodes save significant amount of energy while discovering the desired ratio of contacts. On the other hand, in the mean oracle case, the sizes of searching intervals are proportional to the mean of contact waiting time and contact duration and not affected by whether the node mobility is regular or not. Therefore, the estimated searching intervals are wide even for a small p, and the power management behaves like that of the zero knowlege case for large p. Thus, p does not serve as a good tuning parameter in the mean oracle case when γ is small, i.e., the node mobility is regular.

Figure 17 shows the energy consumption of power management mechanisms as γ increases as in 2, 11, 20, and 89%. As γ increases, more energy is consumed for the same contact discovery ratio because larger searching intervals are required to discover the same amount of contacts due to the increasing randomness in node mobility. Figure 17 also shows that in the same mobility scenario, both mean and mean/variance oracle cases consume similar amount of energy for the same contact discovery ratio because they use similar size of searching intervals for the same contact discovery ratio. However, as γ increases, the range of actual contact discovery ratio of the mean/variance oracle case shrinks because the size of searching intervals cannot be narrowed even for small p when γ is large. As a result, nodes discover contacts not only more than the desired ratio, but also too much due to the wide searching intervals. Thus, p does not serve as a good tuning parameter in the mean/variance oracle case when γ is large, i.e., the node mobility is random. Meanwhile, in the mean oracle case, the range of the discovered contacts spans from about 20% to 100% in all the mbility scenarios. In addition, when γ is 89%, the energy consumption of the mean oracle case is proportional to the actual contact discovery ratio. Because contacts occur randomly, the discovered number of contacts is proportional to the time spent on searching for contacts as well as the energy expended on searching and having contacts.

In summary, the partial knowledge could help nodes to estimate the efficient time ranges to search for contacts, which save energy while discovering contacts. Meanwhile, the performance sensitivity of each power management mechanism depends on mobility characteristics. When the node mobility is regular, nodes could save significant amount of energy by using either mean or mean/variance oracle while discovering more than the desired contact discovery ratio. However, in the mean oracle case, p does not serve as an efficient tuning parameter. To control energy consumption and contact discovery better, we may need an additional mapping from p to another tuning parameter which controls the discovery ratio close to the desired discovery ratio while saving energy. However, in the mean/variance oracle case, p serves as an efficient tuning parameter between energy consumption and contact discovery. Thus, the additional information about variance is useful for the scenarios with regular node mobility. On the other hand, when the node mobility is highly random, p does not serve as a tuning parameter at all in the mean/variance oracle case, while it does in the mean oracle case. Thus, the additional information about variances is useless for the scenarios with highly random node mobility.

3.5.3 Impact of Power Management on Message Delivery

In this section, we evaluate the impact of power management mechanisms on message delivery in two mobility models, MF and RWP, as described in Section 3.5.1. In MF, γ between a ferry and nodes lies in between 11% and 12%. In RWP, γ between any pair of nodes with contacts lies in between 91% and 99%. Thus, the MF scenario represents a regular node mobility and the RWP scenario represents a highly random node mobility.



(a) contact

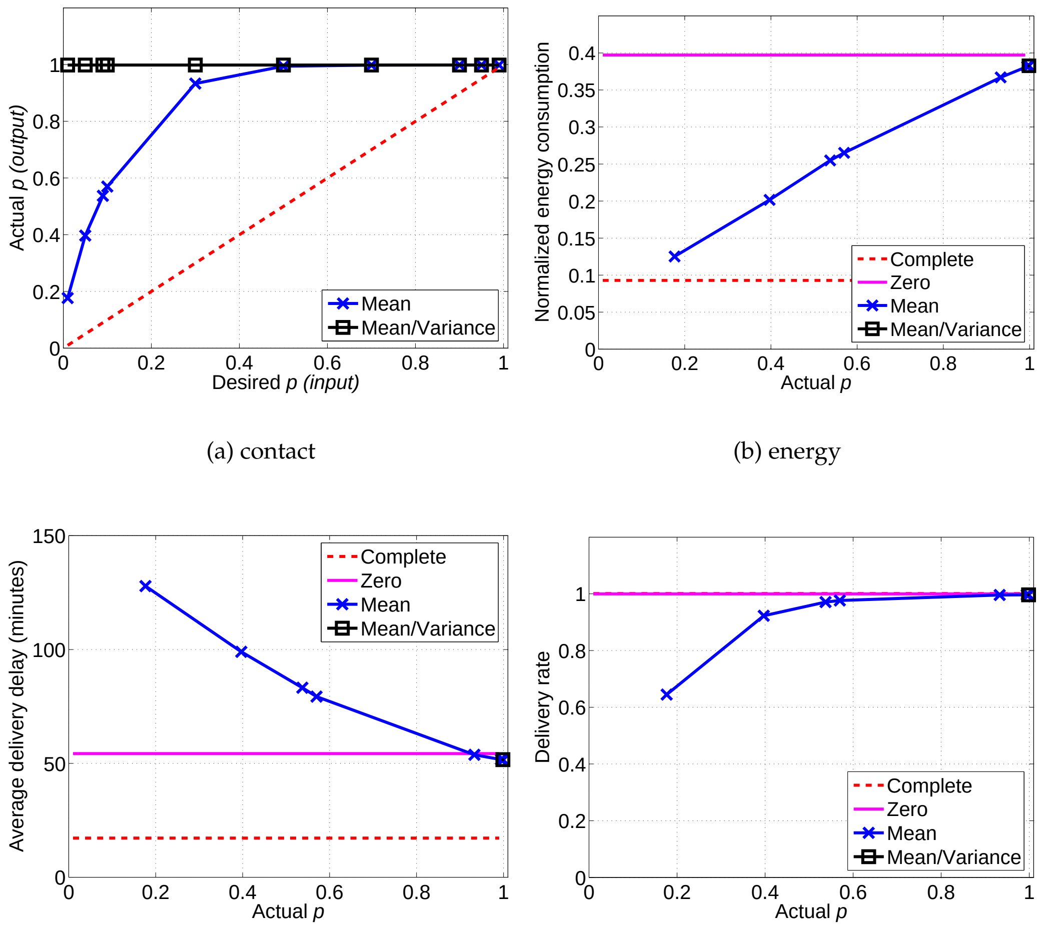
(b) energy

(c) delay

(d) delivery rate

Figure 18: The impact of power management on message delivery (MF)

Figure 18 shows the impact of power management on message delivery in the MF scenario. In Figures 18(a) and (b), we show that actual contact discovery ratio and normalized energy consumption with 10 nodes is similar to the result with two nodes, shown in Figures 16(b) and 17(b). In Figure 18(c), we show the average delivery delay of messages. Both the zero and complete knowledge cases achieve the lowest delivery delay because nodes in these cases discover all the contacts. In the partial knowledge cases, the delay is long for small p, but it decreases as the actual contact discovery ratio p increases because more contacts are discovered. In Figures 18(b) and 18(c), we observe that p trades between energy and delay. Figure 18(d) shows the delivery rate. While the complete and zero knowledge



(a) contact (b) energy

(c) delay (d) delivery rate

Figure 19: The impact of power management on message delivery (RWP)

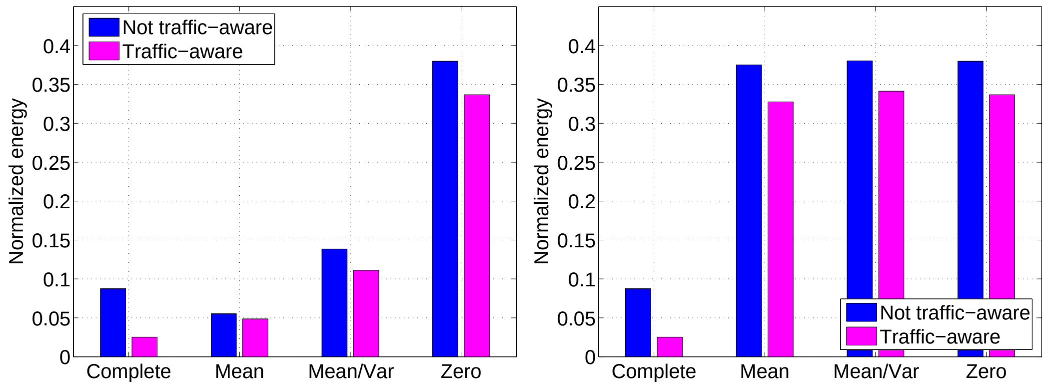
cases deliver most of the messages, the delivery rates of the partial knowledge cases are low for small contact discovery ratio and increases as the discovery ratio increases.

Figure 19 shows the impact of power management on message delivery in the RWP scenario. In Figure 19(a), we show that the actual contact discovery ratio of the mean/variance case is almost 100%. Since γ is large, the node mobility in RWP is highly random. As a result, the size of searching intervals is wide enough that power management behaves as in the zero knowledge case. In the mean oracle case, the contact discovery ratio increases monotonically as p increases. Compared with Figure 16(d), Figure 19(a) also shows that the actual contact discovery of the RWP scenario is greater than that of the MF scenario for the same input p even though γ in RWP is greater than that of MF in Figure 16(d). In RWP, when a node has a contact, it may discover more contacts with other nodes accidentally, which increases the actual contact discovery ratio. In Figure 19(b), we show that the energy consumption of all power management mechanisms is less than 40% of the case without power management. As the actual contact discovery increases, the energy consumption of the mean oracle case increases linearly since more contacts are discovered. In the mean/variance oracle case, nodes consumes energy equivalent to the zero knowledge case since the mean/variance oracle case behaves like the zero knowledge case with wide searching intervals. In Figure 19(c), the complete knowledge case acheives the lowest delay because nodes discover all contacts and know the shortest routing paths for messages. The delivery delay in the zero knowledge case is longer than the complete knowledge case because the lack of information leads messages to travel around before arriving at their destinations. Finally, Figure 19(d) shows that the delivery rates of the mean oracle case is low for small contact discovery ratio and increases as the ratio increases. The other cases deliver most of the messages.

In both MF and RWP scenarios, the delivery rate of the partial knowledge cases is lower than that of the complete knowledge case because long delivery delay causes messages to timeout and to end up being discarded. In our simulation, when the timeout is increased to one day, all mechanisms achieve more than 99% delivery rate, except for the mean oracle case with $p = 0.01$ in MF. Thus, the timeout value can be used as a performance measure as well as a deadline for messages.

3.5.4 Traffic-Aware Enhancement of Power Management

In this section, we evaluate how much energy can be saved by applying traffic-aware enhancement to the contact mode of power management mechanisms in MF. Figure 20 shows the normalized energy consumption when desired contact discovery ratio p is 0.01 and 0.99. When p is 0.01, the mechanisms with partial knowledge intend to discover only 1% of contacts. As a result, nodes discover fewer contacts in the partial knowledge cases than in the case with zero or complete knowlege. Thus, when a node has a contact in a



(a) Desired p=0.01 (MF)

(b) Desired p=0.99 (MF)

Figure 20: The impact of traffic-aware enhancement in the contact mode

partial knowledge case, it may need to utilize most of the contact time to forward accumulated messages, resulting less chance to save energy through traffic-aware enhancement than in the zero or complete knowledge cases. When p is 0.99, the mechanisms with partial knowledge also discover most of contacts. Thus, a node may have less messages to forward within each discovered contact, resulting more chance to save energy through traffic-aware enhancement than when p is 0.01. In Figure 20, each bar represents the energy consumption of four power management mechanisms with or without traffic-aware enhancement as indicated: complete knowledge, partial knowledge with the mean oracle, partial knowledge with the mean and variance oracles, and zero knowledge.

In Figure 20(a), we show the normalized energy consumption of each power management mechanism without or with traffic-aware enhancement when p is 0.01. Figure 20(a) shows that nodes save the largest amount of energy through traffic-aware enhancement in the complete knowledge case. In the complete knowledge case, nodes discover all the contacts, so nodes may have less messages to forward at each contact compared with other cases. Thus, nodes have more remaining time of contacts to sleep after exchanging messages, resulting in more energy savings though traffic-aware enhancement. In the zero knowledge case, although nodes discover all the contacts, the total contact time is less than that in the complete knowledge case because a part of contact time is lost while discovering the contact. Moreover, nodes in the zero knowledge case may forward the same messages multiple times due to its routing feature, resulting in less remaining time of contacts to sleep after exchanging messages. In the partial knowledge cases, nodes discover less contacts than in the complete or zero knowledge cases. So, they need to use more time on each contact to forward accumulated messages, resulting in less energy savings in the partial knowledge cases than in the complete or zero knowledge cases.

In Figure 20(b), we show the normalized energy consumption of each power management mechanism when p is 0.99. Because nodes in the partial knowledge cases discover most of the contacts, they save as much energy as in the zero knowledge case through traffic-aware enhancement. However, their total energy consumption increases significantly compared with the case when p is 0.01 because of extra searching time for contacts.

In Figure 20, we demonstrate that the traffic-aware enhancement in the contact mode save additional energy in all four power management mechanisms regardless of p value. However, greater energy can be saved by the complete knowledge or partial knowledge with low p than by the enhancement in the contact mode. Thus, it is more important to have more knowledge and to estimate the right desired contact discovery ratio p to save energy while achieving the desired performance of message delivery.

3.6 Summary

In this chapter, we investigate power management for a single radio architecture in DTNs. We present a power management framework, in which nodes switch among different power management modes according to the available knowledge about the network connectivities. In addition, we provide an explicit means to trade between energy and delivery performance when the statistical information about waiting time between contacts and contact duration is available. We also devise a mechanism to enhance the performance of our power management using the local traffic information in real time.

Our simulation results from various mobility scenarios show that the efficiency of the trading between energy and delivery performance depends on the network characteristics as well as the deployed mechanisms. Specifically, when partial knowledge is available, more knowledge about network dynamics (i.e., variance) is useful only when its value is relatively small compared to the mean value: in other words, when the node mobility is relatively regular. When the node mobility is random, our mechanism using only mean values discover contacts close to our desired ratio in MF. However, in RWP, it discovers a lot more than the desired ratio of contacts. Also, when the node mobility is intermediate, both mechanisms discover a lot more contacts than the desired ratio in MF, consuming more energy than necessary. Therefore, it is worth to consider different power management approaches for networks with intermediate or high randomness in node mobility. In the next chapter, we consider mechanisms in which radios search for contacts continuously without limiting its search within narrow time intervals in order to suit for the random mobility cases.

CHAPTER IV

HIERARCHICAL POWER MANAGEMENT

4.1 *Introduction*

In the previous chapter, we have shown how nodes can effectively use information about network connectivity to predict when to enable their wireless interfaces and search for contacts. However, those results only applied to a network with a certain degree of regularity, for instance, where a node follows a relatively periodic schedule. Thus, a network with significant randomness in node mobility requires different mechanisms to save energy while delivering messages.

In this chapter, we examine the possibility of using a hierarchical radio architecture in mobile DTNs, in which nodes are equipped with two complementary radios: a long-range, high-power radio and a short-range, low-power radio. In this architecture, energy can be conserved by using the low-power radio to discover contacts with other nodes and then waking up the high-power radio to undertake the data transmission. Also, the low-power radio can be used to search for contacts continuously without limiting its search within narrow time intervals in order to suit for the random mobility cases. Most previous studies using this hierarchical radio architecture have considered only densely deployed networks, in which the short range of the low-power radio is sufficient to discover each other ([64, 68, 55]). However, DTNs are generally applicable to sparser networks, where the low-power radio may reach a subset of other nodes that could be reached by the high-power radio, even when the nodes are mobile. Therefore, if a node relies only on the low-power radio to discover contacts, it may miss them due to the shorter range. To avoid missing contacts, we propose a generalized power management scheme that uses *both* radios to participate in contact discovery. This generalized scheme controls the *wake-up interval* of each radio, which it uses to trade between energy savings and the performance of message delivery. In addition, we devise an adaptive sleeping algorithm that decides *how* to sleep (i.e., turn off, disable, or stay on) based on the expected overhead for a given wake-up interval.

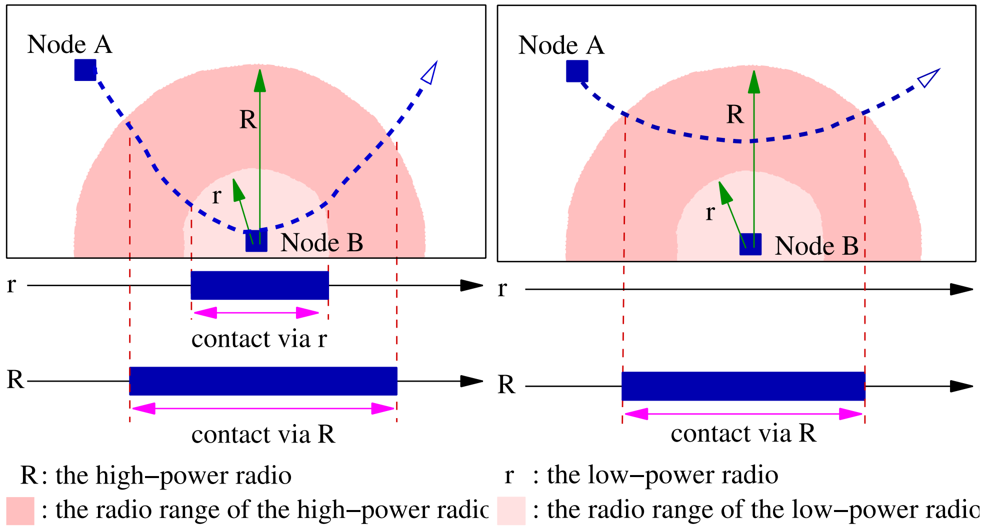
We compare our generalized scheme with two alternative schemes: one that only uses the high-power radio for discovery, and one that only uses the low-power radio. Our evaluation results show that the scheme relying only on the low-power radio achieves the best energy efficiency in discovering contacts, while it may miss some contacts due to the shorter range. On the other hand, the generalized two-radio scheme can tune wake-up intervals of both radios to balance between energy efficiency and delivery performance. However, these gains depend heavily on what the wake-up interval of the radio is set to.

The remainder of this chapter is organized as follows. In Section 4.2, we describe our system model. In Section 4.3, we present our power management framework. Section 4.4 contains an evaluation of our two-radio scheme under various wake-up intervals. We summarize the chapter in Section 4.5.

4.2 System Model

In a mobile DTN, two nodes communicate with one another during *contacts* [30] that occur when the two nodes, either mobile or stationary, are within the radio range of each other. Even if the devices are equipped with multiple radios, the nodes may or may not discover a particular contact opportunity depending on the range of the radios, which radios are active, and the movement trajectory of the two nodes. Figure 21 shows two possible contact scenarios using a two-radio system. In Figure 21(a), node A moves along the trajectory shown and node B stays in one location. Node A first enters within B's high-power radio range and then within B's low-power radio range. As a result, the nodes have a long contact via their high-power radios and a short contact via their low-power radios. In contrast, Figure 21(b) shows that when node A passes node B using a different trajectory, it only enters the range of the high-power radio of node B.

While using a low-power radio alone may discover less contact opportunities than a high-power radio it does so at substantially reduced energy costs [26, 55]. Table 3 shows the power usage of two sample radios, one high-power radio: a DLink 802.11b card, and



(a) Contacts via both radios

(b) A contact only via the high-power radio

Figure 21: Contacts discovered by the high-power and low-power radios, where R and r are the high-power and low-power radios, respectively

Table 3: Power usage of a DLink DCF-660W Wireless CompactFlash 802.11b card and a Chipcon CC2420 RF transceiver (unit:Watt)

Activity	Transmit	Receive	Idle	Doze	Off
DLink	1.4349	1.3832	1.0649	0.2005	0
Chipcon	0.1007	0.0802	0.0718	0.0001	0

one low-power radio: a Chipcon CC2420 radio [3]. The power usage of each wireless interface depends on whether it is *transmitting*, *receiving*, *idling* (when listening to the wireless medium without transmitting nor receiving), *dozing* (when the wireless interface is disabled, but powered), or *off*. When dozing, a node consumes an order of magnitude less power than when idling, while an idling node consumes power at the same order of magnitude as a receiving or transmitting node. In addition, the CC2420 radio consumes an order of magnitude less power than the 802.11 radio for each activity. Thus, it can discover contacts using substantially less energy. However, its outdoor range is limited to 50-100m, while the 802.11 radio has a range of 250-500m.

When a radio is not in use, energy can be saved by turning off or disabling the radio

Table 4: Transition overhead of a DLink DCF-660W Wireless CompactFlash 802.11b card and a Chipcon CC2420 RF transceiver (UNIT: second, joule)

Action	Dlink Air DCF-660W		Chipcon CC2420	
	Latency (s)	Energy (J)	Latency (s)	Energy (J)
Disable	0.0078	0.0104	N/A	N/A
Enable	0.0202	0.0221	N/A	N/A
Turn-off	2.4001	2.3529	0	0
Turn-on	2.5235	2.4738	0.0004	0.0002

(i.e., placing in the doze state). If turned off, the radio does not consume any power, yet the overhead to turn it on is significant. If disabled, the radio has lower overhead to enable, but it still consumes power. Table 4 shows the measured overhead of a DLink DCF-660W Wireless CompactFlash 802.11b card and a Chipcon CC2420 RF transceiver in terms of latency and energy consumption [15]. Both latency and energy consumption to turn off/on a DLink 802.11b card is two orders of magnitude higher than those to disable/enable the card. The overhead to turn on a CC2420 RF transceiver is four orders of magnitude lower than that of a DLink 802.11b card.

4.3 DTN Radio Discovery

In this chapter, we consider a DTN consisting of mobile nodes as well as stationary nodes, which have two radio interfaces: one with a long radio range and high-power, e.g., a 802.11 wireless card, and the other with a short radio range and low-power, e.g., a CC2420 radio. We only account for the communication energy consumption of a wireless interface and do not consider other sources such as computation or mobility. Also, we assume that nodes have no a-priori knowledge about other nodes' mobility and contacts must be discovered by one or both radios of the nodes. To discover contacts, a radio broadcasts messages called *beacons* periodically. To save energy while discovering contacts, a radio has three power management modes: *search*, *contact*, and *dormant*. In the search mode, the radio wakes up periodically to discover a contact. This period is called a *wake-up interval*. In the contact mode, the radio stays awake to exchange messages with other nodes that it previously discovered in the search mode. In the dormant mode, the radio is not used and

Table 5: Power management mechanisms depending on how the radios are used

Power management	low-power radio	high-power radio
CAM	Not used	Always on
PSM	Not used	sleep/wake-up cycling to discover contacts
SPSM	Sleep/wake-up cycling to discover contacts	Awakened by the low-power radio
GPSM	Sleep/wake-up cycling to discover contacts	Sleep/wake-up cycling to discover contacts, *and* Awakened by the low-power radio

remains asleep.

Given that the high-power radio is always necessary for data transfer, there are four possible variations of this general framework. Table 5 summarizes the power management mechanisms used in this chapter. The *Continuous Aware Mechanism (CAM)* uses only the high-power radio of a node. In CAM, the high-power radio always stays awake to search for other nodes. The *Power Saving Mechanism (PSM)* also uses only the high-power radio, however it alternates between sleeping and waking up while discovering contacts. While similar, note that this PSM is a generalization of 802.11's PSM mode. The *Short-range-radio-dependent Power Saving Mechanism (SPSM)* uses both low-power and high-power radios of a node. In this mechanism, the low-power radio alternates between sleeping and waking up to discover contacts, while the high-power radio sleeps and is awakened by the low-power radio only after discovering a contact. Finally, the *Generalized Power Saving Mechanism (GPSM)* uses both the low-power and high-power radios of a node. In this mechanism, both radios alternate between sleeping and waking up to discover contacts. If a contact is discovered by the low-power radio, the high-power radio is awakened. If a contact is discovered by the high-power radio, the radio stays awake as long as it has contact with the other node. In the remainder of this section, we describe each of these mechanisms in detail, with the exception of the rather straightforward CAM.

4.3.1 Power Saving Mechanism (PSM)

In PSM, a node uses only one high-power radio and its radio periodically sleeps and wakes up to save energy while discovering contacts. Such a scheme was first described in our previous work [39] and we summarize it below for completeness.

We describe the wake-up behavior in the search mode using two quantities: *wake-up interval* and *wake-up duration*. At every wake-up interval, the node wakes up for a wake-up duration. During that duration, the node does two things: transmits beacons and listens for other nodes' beacons. In order to avoid transmitting over other nodes' beacons, it uniformly randomly chooses a time within the duration. If it hears another nodes are beaconing, it enters the contact mode. To ensure that a node wakes up when other nodes are beaconing, we restrict wake-up intervals to multiples of one another, starting on discrete intervals synchronized by GPS clocks. Figure 22 illustrates these modes when a node contacts only one node. Initially, the radio of a node is in the search mode since it is not in contact with node k. At the beginning of a wake-up duration, it chooses a random time within the wake-up duration to broadcast a beacon. If it does not receive another node's beacon, the radio sleeps at the end of the wake-up duration and wakes up at the beginning of the next wake-up duration. If the radio receives a beacon, it enters the contact mode and stays awake to exchange messages. In the contact mode it continues to send beacons periodically and listens for beacons from the same node. If the radio fails to receive a certain number of beacons consecutively, it considers the contact is terminated and returns to the search mode. When a node contacts multiple nodes, its radio should use multiple contact states to transit between the power management modes. The detailed procedure and algorithm can be found in [39].

4.3.2 Short-range-radio-dependent Power Saving Mechanism (SPSM)

In SPSM, a node depends only on the low-power radio to discover contacts. As in PSM, nodes have synchronized clocks and both radios have independent wake-up intervals and durations.

This scheme is identical to PSM except that the low-power radio conducts PSM's search

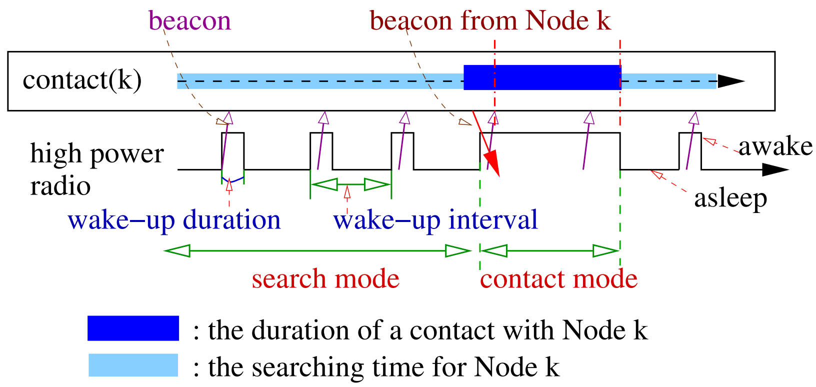


Figure 22: Transition between power management modes when a node has contacts with only one node, node k, using PSM

mode as illustrated in Figure 23. Initially, the high-power radio is in the dormant mode, and the low-power radio is in the search mode. If the low-power radio detects a contact, it wakes up the high-power radio, which then enters the contact mode. In the contact mode, the high-power radio stays awake to exchange messages, using the beacons to detect when the contact ends. At the same time, the high-power radio may detect other contacts by receiving beacons. Meanwhile the low-power radio continues to search for other contacts. When all the contacts terminate, the high-power radio returns to the dormant mode and the low-power radio continues to search.

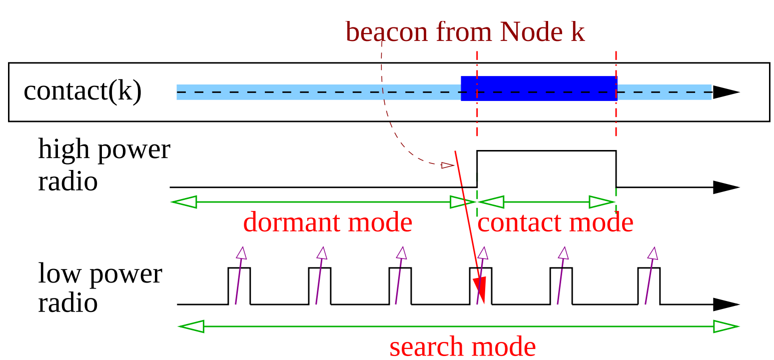


Figure 23: Transition between power management modes when a node has contacts with only one node, node k, using SPSM

4.3.3 Generalized Power Saving Mechanism (GPSM)

In GPSM, a node utilizes both radios to discover contacts. The two radios have separate wake-up intervals and wake-up durations. For instance, the high-power radio may have a larger wake-up interval than the low-power radio, in which case the low-power radio searches for contacts more frequently than the high-power radio.

Figure 24 illustrates the power management with an example scenario. Initially, a node has no contact, so that both of its radios are in the search mode. If the low-power radio receives a beacon from node k, it wakes up the high-power radio and keeps searching for other contacts. If the high-power radio receives a beacon from node k, it enters the contact mode and stays awake to exchange messages. It also sends and listens for beacons from the same node to determine when the contact terminates.

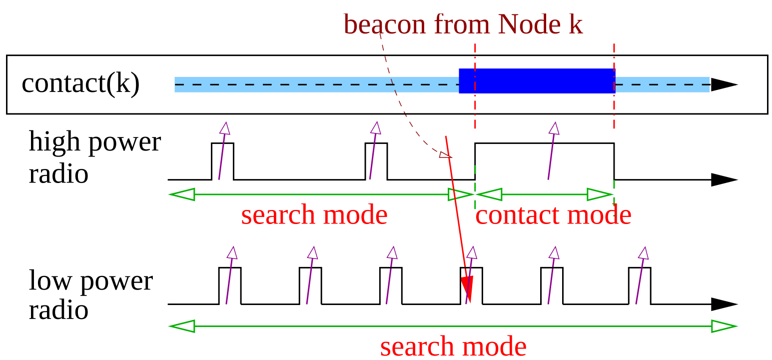


Figure 24: Transition between power management modes when a node has contacts with only one node, node k, using GPSM

This mechanism is a generalized scheme that includes PSM and SPSM at its two extremes. If we set the wake-up interval of the low-power radio to infinity, GPSM is equivalent to PSM. Also, if we set the wake-up interval of the high-power radio to infinity, it is equivalent to SPSM.

4.3.4 Adaptive Sleeping Algorithm

As a radio sleeps and wakes up searching for contacts, a node can choose to turn off the radio completely, disable it, or even leave it awake. The choice depends on the chosen wake-up interval, and the transition costs of the particular radio. While the node can save energy by placing the node in a low-power state, the energy cost of that transition may not justify the state-change. Further, the wake-up interval may be so short that the latency in switching states does not permit the state-change. As we show in Algorithm 1, the break-even points for each possible state can be computed by balancing the transition costs with

Table 6: The range of sleeping time s to be in each state (unit: seconds)

State	Off	Doze	Awake
DLink	s > 23.94	23.94 ≥ s > 0.03	0.03 ≥ s
CC2420	s > 0.01	N/A	0.01 ≥ s

the energy saved and the permissible transition times.

Algorithm 1 Adaptive Sleep

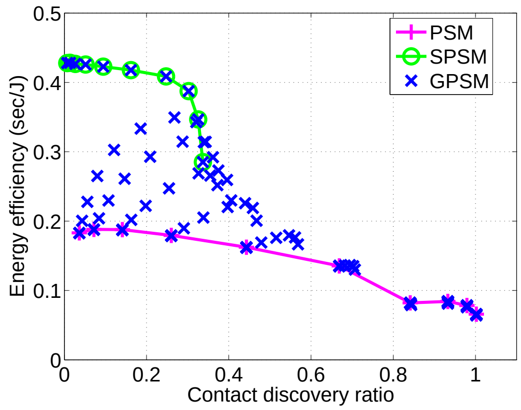
Let L_{state} = Latency to enter and leave the *state*
Let $E_{state}(s)$ = Energy consumption in the state for time $(s - L_{state})$ plus energy to enter and leave the state

Input: sleeping time s
function AdaptiveSleep(s)
if $s > L_{off}$ and $E_{off}(s) < \max(E_{doze}(s), E_{idle}(s))$ **then**
  turn off the radio
  schedule to wake it up after (s − latencyToTurnOn)
else if $s > L_{doze}$ and $E_{doze}(s) < E_{idle}(s)$ **then**
  disable the radio
  schedule to enable it after (s − latencyToEnable)
else
  idle for s
end if

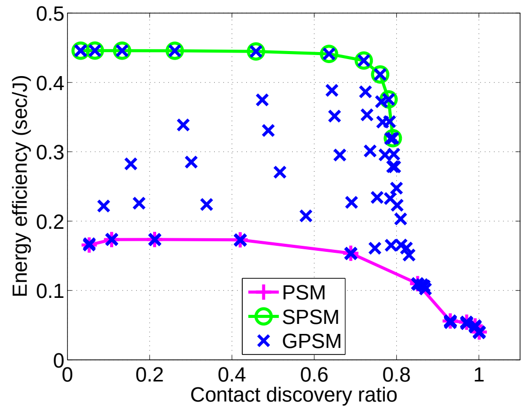
As an example, we use the power consumption and transition overhead of a DLink DCF-660W Wireless CompactFlash 802.11b card and a CC2420 RF transceiver as shown in Tables 3 and 4 [15]. Table 6 shows the range of sleeping times for each radio to enter each state. For instance, if the wake-up interval is greater than 23.94 seconds, the 802.11b card should enter the off state, as opposed to the doze state.

4.4 *Performance Comparison*

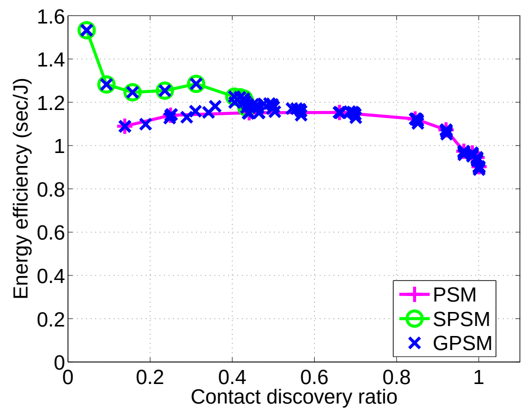
To provide some insight into the relative performance of these schemes, we provide a set of simulation results. Specifically, we investigate the impact of the wake-up interval of the high-power radio on the contact discovery performance. We consider two metrics: (1) *contact discovery ratio*, which is the discovered contact time divided by the total contact time that the high-power radio could have discovered when using CAM, and (2) *energy efficiency*, which is the average amount of discovered contact time per unit energy used. In other words, we measure how much energy the system spends to find a certain amount of



(a) RWP



(b) MF



(c) UMassDieselNet

Figure 25: The impact of wake-up intervals to energy efficiency and contact discovery ratio of power management schemes under the various mobility scenarios

message transfer time—the more transfer time discovered per joule of energy, the higher the efficiency is.

We use the ns-2 simulator with the following parameters [2]. We use the 802.11 MAC layer for both interfaces at different channels. Also, both interfaces set the beacon period to 2 seconds and the wake-up duration to 300ms. Each simulation runs for 10 days of simulated time and each point of our graphs represents the average of ten runs. For the experiment we use the energy consumption of an 802.11 and a CC2420 radio as shown in Table 3. For the low-power radio, the radio range and the data rate are 100m and 76.8Kb/s, respectively. For the high-power radio, the radio range and the data rate are 250m and

2Mb/s, respectively.

We show results using several different mobility models, i.e., Random-Waypoint (RWP) [34], Message Ferrying (MF) [78], and UMassDieselNet scenarios [4]. In RWP 20 nodes move around in 10km×10km area with random speeds between 5m/s and 10m/s[1] and exponential random pause times with an average of 30 seconds. In MF the network consists of nine stationary nodes and a mobile node, called a ferry. We place stationary nodes in a grid of 10km×10km area sparsely. Then the ferry visits each node in order and repeats the route. Whenever the ferry reaches a node, it selects a random pause time and a random speed as a mobile node in the RWP scenario. In the UMassDieselNet scenario, the network consists of 10 buses moving around in 20km×25km and one stationary node that uses our power management mechanisms. The UMassDieselNet scenario is a trace-driven simulation using mobility traces from a live bus transit system. The stationary node is placed to improve the capacity of networking among buses, and its location is calculated by using a greedy algorithm [81]. We assume that buses have unlimited energy, e.g., from vehicular batteries, so that their radios are always on. This UMassDieselNet scenario demonstrates the use of our power management in real implementation to conserve energy. The key differences among mobility scenarios are (a) the distance that nodes pass one another: in the RWP and UMassDieselNet scenarios, contacts may occur at much longer distances than in the MF scenario, and (b) the frequency that nodes contact each other: in the UMassDieselNet scenario, due to the placement of the node in a high-traffic area, the stationary node has frequent contacts, with a small inter-contact time.

In Figure 25, we show the impact of the wake-up intervals of the radios on the performance of contact discovery and energy efficiency of three power management schemes under the three mobility scenarios. We use a set of wake-up interval candidates in second as follows: $\{2, 4, 8, \cdots, 1024, \text{and } \infty\}$. For PSM, the wake-up interval of the high-power radio is set to one of the finite candidates. For SPSM, that of the low-power radio is set to one of the finite candidates. For GPSM, the wake-up interval of each radio is set to one of the candidates while only one radio can be set to ∞ at once. For each selection of wake-up

[1]We use a non-zero minimum speed to adjust the stability problem of RWP as described in [76].

intervals, x-axis depicts the contact discovery ratio ranging from zero to one, while y-axis depicts the corresponding energy efficiency.

Figure 25(a) shows the results in the RWP scenario. The graph shows that SPSM achieves the highest energy efficiency for the wake-up intervals. However, SPSM can discover contact time only up to 38% due to its short radio range. In the meantime, PSM can discover up to 100% of contact time. However, it has the lowest energy efficiency. Since GPSM includes PSM and SPSM at its two extremes, it has wake-up intervals that achieve the highest energy efficiency for all contact discovery ratio.

Figure 25(b) shows the results in the MF scenario, which is similar to the RWP scenario. The only difference is that SPSM can discover up to 80% of contact time in this scenario because the ferry always enters the radio range of the low-power radio whenever it has a contact via the high-power radio.

Finally, Figure 25(c) shows the results in the UMassDieselNet scenario. Compared to the previous two scenarios, the relative energy efficiency of SPSM to that of PSM is much smaller in the UMassDieselNet scenario. Because of the placement of the stationary node in a high-traffic area, the node has frequent contacts with a small inter-contact time. As a result, the node spends much shorter time in searching than nodes in the RWP and MF scenarios.

4.5 Summary

In this chapter we investigate power management in DTNs with high randomness in the node mobility. We present a hierarchical power management framework, in which nodes control two radio interfaces to discover contacts. Our simulation results from three mobility models show that our generalized power management mechanism balance energy efficiency and contact discovery by tuning the wake-up intervals of the two radios. Also, our experiment demonstrates that the wake-up intervals of the radios greatly impact energy efficiency and contact discovery performance, and the usefulness of the second radio is highly dependent on those intervals and mobility scenarios. More importantly, this does not answer the question as to how to set the interval in the first place. In the next chapter, we provide this as an analytical result when additional information about contacts and traffic load is available.

CHAPTER V

TRAFFIC-AWARE OPTIMIZATION IN HIERARCHICAL POWER MANAGEMENT

5.1 *Introduction*

In the hierarchical power management mechanisms proposed in Chapter 4, wake-up intervals of each radio affect the energy savings and contact discovery greatly. However, the fact does not answer the question as to how to set the intervals in the first place. In this work, we incorporate the knowledge of traffic load and network topology changes to determine the optimal wake-up intervals(i.e., sleep/wake-up cycling). When the traffic load can be estimated, nodes do not need to discover all the contacts. They may discover just enough contacts to deliver the traffic load. Thus, we devise approximation algorithms that minimize the overall energy consumption, while discovering enough contacts to handle the expected traffic load.

We evaluate our schemes through simulations and compare them against single radio architectures, and against mechanisms that do not incorporate information about the load. The simulation results show that our approximation algorithm could reduce energy consumption from 60% to 99% compared with the case without power management. In addition, by evaluating power management schiemes in three different mobility scenarios, we show that our generalized power management mechanism balance energy efficiency and delivery performance by tuning the wake-up intervals of the two radios. Also, the relative energy efficiency of using the additional low-power radio increases as the nodes' searching time increases. Finally, we show that the usefulness of the low-power radio depends on the effective radio range of the high-power radio, in which high-power radios can achieve high throughput: as the effective radio range of the high-power radio decreases, the additional low-power radio becomes more useful.

The remainder of this chapter is organized as follows. In Section 5.2, we propose wake-up interval estimation for our power management according to the expected traffic load. Section 5.3 presents an evaluation of our traffic-aware power management schemes. We summarize the work in Section 5.4.

5.2 *Traffic-Aware Wake-Up Interval Estimation*

The critical issue in all of the power management mechanisms is determining the proper wake-up intervals. In each mechanism, the wake-up intervals can be used to trade between energy and delivery performance. Specifically, when the wake-up intervals are long, energy can be saved at the cost of missing contacts, which in turn results in poor delivery performance. That is, if the traffic generation rate is greater than the delivery rate, the node will eventually drop packets due to its limited buffer space. To avoid dropping packets, we state the problem as follows: *For each node, find the wake-up intervals that leads to discovering enough contacts to deliver the traffic load while minimizing energy consumption.*

To formulate the problem, we assume that statistical information about contacts and traffic load between each pair of nodes is available. This information is often already available in a DTN: nodes observe and exchange the history of contacts and traffic load to make efficient routing decisions ([18, 30]).

To address the problem, we define the *contact arrival rate* as the number of contacts between two nodes over a unit time and *expected bandwidth* as the maximum amount of data that can be delivered by the discovered contacts between two nodes over a unit time. With these definitions, we can estimate the expected bandwidth for given wake-up intervals using contact duration and contact arrival rate per pair of nodes. Unfortunately, the distribution of inter-arrival times and contact durations is generally not known, especially for many common mobility models [14]. Without this, it is not possible to develop general, optimal algorithms. Therefore, we devise approximation algorithms assuming that contacts arrive according to a Poisson process and contact durations are constant and equal to the average contact duration from a mobility scenario. In Section 5.3, we validate this approximation and show that it works well in practice. The notation used in the wake-up

Table 7: Notation used in the wake-up interval estimation, where the subscript ij indicates that the parameter is specific to the link between node i and node j

N	the set of nodes in a network
R	the radio range of the high-power interface
r	the radio range of the low-power interface
θ	the bandwidth of the high-power interface
f_{ij}	the traffic load on the link (i, j)
w^h, w^l	the wake-up intervals of the high-power and low-power radios, respectively
$\lambda_{ij}^h, \lambda_{ij}^l$	the contact arrival rates when R is used and when r is used, respectively
d_{ij}^h, d_{ij}^l	the contact duration when R is used and when r is used, respectively
$c_{ij}^{psm}(w^h)$	the expected amount of a contact that can be discovered by PSM for a given w^h
$c_{ij}^{spsm}(w^l)$	the expected amount of a contact that can be discovered by SPSM for a given w^l
$c_{ij}^{gpsm}(w^h, w^l)$	the expected amount of a contact that can be discovered by GPSM for given (w^h, w^l)

interval estimation is summarized in Table 7.

5.2.1 Wake-Up Interval Estimation for PSM

Recall that PSM uses only the high-power radio to discover contacts. Since individual nodes have different traffic loads, we choose a different optimal wake-up interval for each node. However, given that nodes wake up at different rates, we must ensure that the nodes can still synchronize their wake-up's so that one node hears another node's beacons. To do this, we choose the wake-up intervals from a set that are a multiple, or even divisor of every other node's wake-up intervals as follows:

$$W^h = \{w^h \mid w^h = 2^k b^h,\ k = 0, 1, 2, ..K\}, \tag{14}$$

where b^h is a unit time period for the high-power radio to send beacons and K is a non-negative integer.

From W^h, we determine the optimal wake-up interval of the high-power radio for node i as follows. First, we estimate the expected bandwidth on the link (i, j) for each wake-up interval $w^h \in W^h$, as illustrated in Figure 26. Initially, an opportunity to have a contact on the link (i, j) starts at time t after the termination of the last wake-up. Because we assume that contacts arrive as a Poisson process, the arrival time of a contact is approximately uniformly distributed within a wake-up interval, i.e., $t \sim \text{uniform}(0,\ w^h)$. At the next wake-up, the contact can be discovered if the contact is longer than the remaining time until the next wake-up, i.e., $w^h - t < d^h_{ij}$, where d^h_{ij} is the contact duration on the link (i, j) that can be discovered by the high-power radio. In addition, the length of the discovered contact time is the contact duration less the remaining time until the next wake-up, i.e., $d^h_{ij} - (w^h - t)$. Thus, the expected discovered contact time $c^{psm}_{ij}(w^h)$ of a contact for a given w^h is calculated as follows. If $w^h \leq d^h_{ij}$,

$$c^{psm}_{ij}(w^h) = \int_0^{w^h} \frac{1}{w^h}(d^h_{ij} - w^h + t)\ \mathrm{d}t. \tag{15}$$

Otherwise (i.e., if $w^h > d^h_{ij}$),

$$\begin{aligned} c^{psm}_{ij}(w^h) &= \int_0^{w^h - d^h_{ij}} \frac{1}{w^h} \cdot 0\ \mathrm{d}t \\ &+ \int_{w^h - d^h_{ij}}^{w^h} \frac{1}{w^h}(d^h_{ij} - w^h + t)\ \mathrm{d}t. \end{aligned} \tag{16}$$

The first integral corresponds to the case when the contact is shorter than the wake-up interval, and the contact opportunity does not overlap with any of the beacon windows. The second case corresponds to the case when the contact duration can be discovered by a wake-up, but the node misses the first part of the contact. As a result,

$$c^{psm}_{ij}(w^h) = \begin{cases} d^h_{ij} - \frac{1}{2}w^h & \text{if } w^h \leq d^h_{ij} \\ \frac{(d^h_{ij})^2}{2w^h} & \text{otherwise} \end{cases} \tag{17}$$

Then, the expected bandwidth on the link (i, j) is $\lambda^h_{ij} \cdot \theta \cdot c^{psm}_{ij}(w^h)$ for a given w^h, where λ^h_{ij} is the contact arrival rate when the high-power radio is used on the link (i, j), and θ is the bandwidth of the high-power radio.

Next, we find the set of wake-up intervals, w^h, that provide bandwidth greater than or equal to the traffic load on the link (i, j). We choose the wake-up interval from this set, by finding the interval that consumes the least amount of energy. In the case of PSM, this is

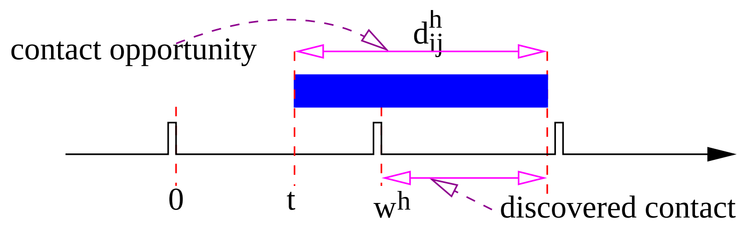


Figure 26: A discovered contact time by PSM for a given w^h

straightforward as it is the maximum wake-up interval w_{ij}^h in the set, i.e.,

$$w_{ij}^h = \max\left\{w^h \mid \lambda_{ij}^h \cdot \theta \cdot c_{ij}^{psm}(w^h) \geq f_{ij},\ w^h \in W^h\right\}, \tag{18}$$

where f_{ij} is the traffic load on the link (i,j). Then, the resulting w_{ij}^h is the wake-up interval that will consume the least amount of energy while satisfying the bandwidth constraint on the link (i,j).

Since the node contacts multiple other nodes, and it does not know which node it will discover next, it must choose the minimum wake-up interval among all of the wake-up intervals it computed for each outgoing link, w_{ij}^h, i.e.,

$$w_i^h = \min\{w_{ij}^h \mid j \in N - \{i\}\}, \tag{19}$$

where N is the set of nodes in the network. Then, the resulting w_i^h is the wake-up interval that consumes the least amount of energy while satisfying the bandwidth constraints for all the links of node i. This result is optimal for the set of discrete wake-up intervals the algorithm considers.

5.2.2 Wake-Up Interval Estimation for SPSM

In contrast with PSM, SPSM relies solely on the low-power radio to discover contacts. As shown in Figure 21, the length of a contact via the low-power radio is shorter than that via the high-power radio. Thus, once a contact is discovered via the low-power radio, the high-power radio is awakened and extends the contact time beyond the time the node leaves the range of the low-power radio. This extended amount of a contact is denoted as δ_{ij} for the link (i,j). This δ_{ij} can be observed by having a node maintain its contact states by both radios and estimate the time difference between the termination of their contacts, i.e., (the termination of a contact discovered by the high-power radio - that of the corresponding contact by the low-power radio). This extension makes the estimation of the expected bandwidth of SPSM different from that of PSM. With this exception, the wake-up interval estimation procedure of SPSM is identical to that of PSM.

Figure 27 illustrates an example scenario to estimate the expected bandwidth on the link (i, j). Initially, a contact starts at time t after the termination of the last wake-up. At the next wake-up, the contact can be discovered if the contact via the low-power radio is longer than the remaining time until the next wake-up, i.e., $w^l - t < d^l_{ij}$, where d^l_{ij} is the contact duration on the link (i, j) when the low-power radio is used. Then, the high-power radio will start having a contact after the transition latency, denoted by L. In addition, the contact will be extended by δ_{ij} by the long range of the high-power radio. As a result, the discovered amount of a contact is the summation of the contact duration by the low-power radio and the extended contact time by the high-power radio less (the remaining time until the next wake-up and the latency to turn on the high-power radio), i.e., $d^l_{ij} + \delta - \{(w^l - t) + L\}$. Here we assume $L \leq \delta_{ij}$ without losing much generality. Thus, the expected discovered contact time of a contact $c^{spsm}_{ij}(w^l)$ is calculated as follows: If $w^l \leq d^l_{ij}$,

$$c^{spsm}_{ij}(w^l) = \int_0^{w^l} \frac{1}{w^l}(d^l_{ij} + \delta_{ij} - w^l + t - L)\,\mathrm{d}t. \tag{20}$$

Otherwise (i.e., if $w^l > d^l_{ij}$),

$$\begin{aligned} c^{spsm}_{ij}(w^l) &= \int_0^{w^l - d^l_{ij}} \frac{1}{w^l} \cdot 0\,\mathrm{d}t \\ &+ \int_{w^l - d^l_{ij}}^{w^l} \frac{1}{w^l}(d^l_{ij} + \delta_{ij} - w^l + t - L)\,\mathrm{d}t \end{aligned} \tag{21}$$

As a result,

$$c^{spsm}_{ij}(w^l) = \begin{cases} d^l_{ij} + \delta_{ij} - L - \frac{1}{2}w^l & \text{if } w^l \leq d^l_{ij} \\ \frac{d^l_{ij}(d^l_{ij} + 2\delta_{ij} - 2L)}{2w^l} & \text{otherwise.} \end{cases} \tag{22}$$

Then, the expected bandwidth between nodes i and j is $\lambda^l_{ij} \cdot \theta \cdot c^{spsm}_{ij}(w^l)$ for a given w^l, where λ^l_{ij} is the contact arrival rate on the link (i, j) when the low-power radio is used. The rest of the procedure to determine the optimal wake-up interval is omitted because it is identical to that of PSM.

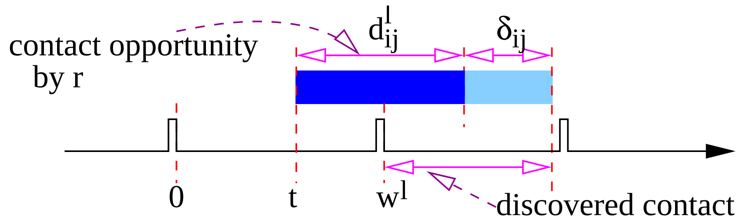


Figure 27: A discovered contact time by SPSM for a given w^l

5.2.3 Wake-Up Interval Estimation for GPSM

This power management uses both high-power and low-power radios to discover contacts. Since contacts can be discovered by either of the radios, the wake-up interval estimation procedure is more complicated than that of the other mechanisms. In brief, we determine the wake-up intervals for node i as follows. First, we estimate the expected bandwidth on the link (i, j) for each pair of wake-up intervals (w^h, w^l) in a given set of discrete wake-up interval pairs: i.e., $\{(w^h, w^l) \mid w^h \in W^h \cup \{\infty\}, w^l \in W^l \cup \{\infty\}\}$, where W^h and W^l are sets of finite discrete wake-up interval candidates provided by a network operator for the high-power and low-power radios, respectively, as in Section 5.2.1. Second, we find a set S_{ij} of wake-up interval pairs that provide enough bandwidth to consume the expected traffic load on the link (i, j) for each $j \in N - \{i\}$. Third, we find the pair of wake-up intervals (w^h_{ij}, w^l_{ij}) in S_{ij} that will consume the least amount of energy assuming that node i contacts only node j, so both nodes choose the same pair of wake-up intervals that satisfies the bandwidth requirement on their link and possibly consumes the least amount of energy. Finally, the node chooses the minimum wake-up intervals w^h_i and w^l_i among w^h_{ij} and w^l_{ij}, respectively. Then, the resulting (w^h_i, w^l_i) is the pair of wake-up intervals that consumes approximately the least amount of energy while satisfying the bandwidth constraints for all the links of node i.

The detailed procedure is as follows. To estimate the expected bandwidth per link, we first calculate the expected discovered contact time of a contact, in which a pair of nodes can communicate with each other through their high-power radios. If either w^l or w^h is infinite, the discovered contact time is c^{psm} or c^{spsm}, respectively. Otherwise, it is calculated as follows. Among the contacts that can be discovered by the high-power radios, some of them can also be discovered by the low-power radio and the others cannot, as shown in Figure 21. To distinguish these two classes in an equation, we denote α_{ij} as the number of contacts that can be discovered by both radios divided by the number of contacts that can be discovered by the high-power radio in CAM. Then, the expected discovered contact time $c_{ij}^{gpsm}(w^h, w^l)$ can be stated as follows:

$$c_{ij}^{gpsm}(w^h, w^l) = \alpha_{ij}\bar{c}_{ij}^{gpsm}(w^h, w^l) + (1 - \alpha)c_{ij}^{psm}(w^h), \tag{23}$$

where $\bar{c}_{ij}^{gpsm}(w^h, w^l)$ is the expected contact time that can be discovered by both radios on the link (i, j) for given (w^h, w^l), and $c_{ij}^{psm}(w^h)$ is the expected contact time that can be discovered by the high-power radio, as defined before.

The contacts that can be discovered by both radios, $\bar{c}_{ij}^{gpsm}(w^h, w^l)$, can also be categorized into subclasses because each radio has a different probability to discover a contact depending on its choice of a wake-up interval. Thus, we categorize them into three classes: (1) contacts that *will* be discovered by both radios, (2) contacts that can be discovered by the high-power radio, but not by the low-power radio, and (3) contacts that can be discovered by the low-power radio, but not by the high-power radio.[1] To represent these classes in an equation, we denote p_{ij}^p as the probability to discover a contact by the high-power radio and p_{ij}^s as the probability to discover a contact by the low-power radio on the link (i, j). Because we assume that the contact durations are constant and contacts arrive as a Poisson process, the arrival time of contacts is approximately uniformly distributed within a wake-up interval. Thus,

$$p_{ij}^p = \begin{cases} \frac{d_{ij}^h}{w^h} & \text{if } d_{ij}^h \leq w^h \\ 1 & \text{otherwise.} \end{cases} \tag{24}$$

Similarly,

$$p_{ij}^s = \begin{cases} \frac{d_{ij}^l}{w^l} & \text{if } d_{ij}^l \leq w^l \\ 1 & \text{otherwise.} \end{cases} \tag{25}$$

[1]Multiple contacts via the low-power radio can occur within a contact via the high-power radio. However, for simplicity, we assume that there is one contact via the low-power radio that corresponds to one via the high-power radio. This assumption can cause moderate error at the estimation of discovered contact time for a given pair of wake-up intervals as shown in Figure 29.

Thus, $\bar{c}_{ij}^{gpsm}(w^h, w^l)$ is the summation of these three classes: i.e.,

$$\begin{aligned}\bar{c}_{ij}^{gpsm}(w^h, w^l) &= \hat{c}_{ij}^{gpsm}(w^h, w^l) \cdot p_{ij}^p \cdot p_{ij}^s \\ &+ c_{ij}^{psm}(w^h) \cdot (1 - p_{ij}^s) \\ &+ c_{ij}^{spsm}(w^l) \cdot (1 - p_{ij}^p), \end{aligned} \tag{26}$$

where $\hat{c}_{ij}^{gpsm}(w^h, w^l)$ is the expected amount of a contact that will be discovered by both radios for given (w^h, w^l), and $c_{ij}^{psm}(w^h)$ and $c_{ij}^{spsm}(w^l)$ are defined as before.

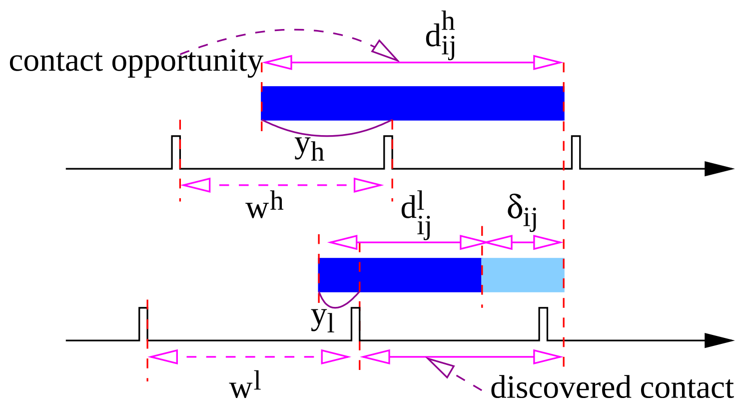


Figure 28: A discovered contact time by both radios using GPSM for a given (w^h, w^l)

The expected amount of a contact that will be discovered by both radios, $\hat{c}_{ij}^{gpsm}(w^h, w^l)$, is illustrated in Figure 28. Initially, a contact opportunity via the high-power radio starts y_h time before the wake-up of the high-power radio, and the corresponding contact opportunity via the low-power radio starts y_l time before the wake-up of the low-power radio. Since we assume that this contact will be discovered by both radios, y_h should be between 0 and $\min(w^h, d_{ij}^h)$. Thus, $y_h \sim \text{uniform}(0, \min(w^h, d_{ij}^h))$. Similarly, $y_l \sim \text{uniform}(0, \min(w^l, d_{ij}^l))$. In addition, if the discovered contact by the high-power radio is longer than that by the low-power radio, i.e., $d_{ij}^h - y_h > (d_{ij}^l + \delta_{ij} - L) - y_l$, the contact will be discovered by the high-power radio first and its discovered length is $d_{ij}^h - y_h$, where L denotes the transition latency of the high-power radio to wake up when w^h is used as its wake-up interval. Otherwise, it will be discovered by the low-power radio first and its discovered length is $d_{ij}^l + \delta_{ij} - L - y_l$. Thus, the expected contact time of a contact that will be discovered by both radios is calculated as follows: If $w^h \leq d^h_{ij}$ and $w^l \leq d^l_{ij}$,

$$\widehat{c}^{gpsm}_{ij}(w^h, w^l) = \int_0^{w^l} \int_0^{d^h_{ij}-d^l_{ij}-\delta_{ij}+L+y_l} \frac{(d^h_{ij}-y_h)}{w^h w^l} \mathrm{d}y_h \mathrm{d}y_l + \int_0^{w^l} \int_{d^h_{ij}-d^l_{ij}-\delta_{ij}+L+y_l}^{w^h} \frac{(d^l_{ij}+\delta_{ij}-L-y_l)}{w^h w^l} \mathrm{d}y_h \mathrm{d}y_l. \tag{27}$$

If $w^h \leq d^h_{ij}$ and $w^l > d^l_{ij}$,

$$\widehat{c}^{gpsm}_{ij}(w^h, w^l) = \int_0^{d^l_{ij}} \int_0^{d^h_{ij}-d^l_{ij}-\delta_{ij}+L+y_l} \frac{(d^h_{ij}-y_h)}{w^h d^l_{ij}} \mathrm{d}y_h \mathrm{d}y_l + \int_0^{d^l_{ij}} \int_{d^h_{ij}-d^l_{ij}-\delta_{ij}+L+y_l}^{w^h} \frac{(d^l_{ij}+\delta_{ij}-L-y_l)}{w^h d^l_{ij}} \mathrm{d}y_h \mathrm{d}y_l. \tag{28}$$

If $w^h > d^h_{ij}$ and $w^l \leq d^l_{ij}$,

$$\widehat{c}^{gpsm}_{ij}(w^h, w^l) = \int_0^{w^l} \int_0^{d^h_{ij}-d^l_{ij}-\delta_{ij}+L+y_l} \frac{(d^h_{ij}-y_h)}{d^h_{ij} w^l} \mathrm{d}y_h \mathrm{d}y_l + \int_0^{w^l} \int_{d^h_{ij}-d^l_{ij}-\delta_{ij}+L+y_l}^{d^h_{ij}} \frac{(d^l_{ij}+\delta_{ij}-L-y_l)}{d^h_{ij} w^l} \mathrm{d}y_h \mathrm{d}y_l. \tag{29}$$

Otherwise (i.e., if $w^h > d^h_{ij}$ and $w^l > d^l_{ij}$),

$$\widehat{c}^{gpsm}_{ij}(w^h, w^l) = \int_0^{d^l_{ij}} \int_0^{d^h_{ij}-d^l_{ij}-\delta_{ij}+L+y_l} \frac{(d^h_{ij}-y_h)}{d^h_{ij} d^l_{ij}} \mathrm{d}y_h \mathrm{d}y_l + \int_0^{d^l_{ij}} \int_{d^h_{ij}-d^l_{ij}-\delta_{ij}+L+y_l}^{d^h_{ij}} \frac{(d^l_{ij}+\delta_{ij}-L-y_l)}{d^h_{ij} d^l_{ij}} \mathrm{d}y_h \mathrm{d}y_l. \tag{30}$$

Next, we find the set S_{ij} of wake-up interval pairs that provide more than or equal to the required bandwidth on the link (i, j) as follows:

$$S_{ij} = \{(w^h, w^l) \mid \lambda^h_{ij} \cdot \theta \cdot c^{gpsm}_{ij}(w^h, w^l) \geq f_{ij}\}, \tag{31}$$

where f_{ij} and λ_{ij} are the traffic load and the contact arrival rate via the high-power radio on the link (i, j), respectively.

Third, we find the pair of wake-up intervals (w^h_{ij}, w^l_{ij}) in S_{ij} that will consume the least amount of energy assuming that node i contacts only node j, so both nodes choose the same pair of wake-up intervals that satisfies the bandwidth constraint on their link and possibly consumes the least amount of energy. In previous mechanisms, the maximum wake-up interval consumes the least energy among candidates. However, in GPSM, it is not obvious whether increasing one wake-up interval will result in energy savings if the other wake-up interval has to be decreased. Thus, we estimate the expected energy consumption for each pair of wake-up intervals in S_{ij} to find the pair that minimizes the energy consumption. To represent the energy consumption of GPSM, the transmission, reception, and idle power of the high-power radio are denoted as P_h^t, P_h^r, and P_h^i, respectively. Also, for a given w^h, the corresponding sleeping power, latency and energy overhead to wake up the high-power radio are denoted as $P_h^s(w^h)$, $L_h(w^h)$ and $\theta_h(w^h)$, respectively. Similarly, those of the low-power radio are denoted as P_l^t, P_l^r, P_l^i, $P_l^s(w^l)$, $L_l(w^l)$ and $\theta_l(w^l)$, respectively. Also, a beacon window of the high-power radio is denoted as a_h, and that of the low-power radio is denoted as a_l. Then, the energy consumption of GPSM is the summation of energy consumption by the low-power radio in the search mode, that of the high-power radio in the searching mode, that of the high-power radio in the contact mode, and the overhead to wake up the high-power radio when the low-power radio discovers contacts. We ignore the transmission energy of beacons because the size of a beacon is so small that its additional energy consumption for transmitting beyond idling is negligible. Then, the energy consumption by the low-power radio in the search mode is

$$A(w^l) = P_l^i \frac{a_l}{w^l} + \theta_l(w^l)\frac{1}{w^l} + P_l^s(w^l)(1 - \frac{a_l + L_l(w^l)}{w^l}) \tag{32}$$

for a unit time. Similarly, the energy consumption by the high-power radio in the search mode is

$$B(w^h) = P_h^i \frac{a_h}{w^h} + \theta_h(w^h)\frac{1}{w^h} + P_h^s(w^h)(1 - \frac{a_h + L_h(w^h)}{w^h}) \tag{33}$$

for a unit time. Since the high-power radio is either searching or having contacts, its energy consumption for searching is

$$C(w^h) = (1 - \lambda_{ij}^h c_{ij}^{gpsm}(w^h, w^l)) \times B(w^h) \tag{34}$$

and that for contacts is

$$D(w^h) = \lambda_{ij}^h c_{ij}^{gpsm}(w^h, w^l) P_h^i \tag{35}$$

for a unit time. Finally, the energy overhead to wake up the high-power radio is

$$H(w^h) = \lambda_{ij}^l p_{ij}^s \theta_h(w^h) \tag{36}$$

assuming that the low-power radio always discovers contacts first. As a result, the energy consumption $E(w^h, w^l)$ of GPSM for given (w^h, w^l) in a unit time is estimated as follows:

$$\begin{cases} C(w^h) + D(w^h) & \text{if } w^l = \infty \\ A(w^l) + D(w^h) + H(w^h) & \text{if } w^h = \infty \\ A(w^l) + C(w^h) + D(w^h) + H(w^h) & \text{otherwise.} \end{cases}$$

When neither wake-up intervals infinite, the energy overhead to wake up the high-power radio can be calculated similarly to the estimation procedure of the discovered contact time. However, its difference from $H(w^h)$ is negligible. Therefore, for simplicity, we use $H(w^h)$ for the energy overhead to wake up the high-power radio even when contacts can be discovered by either radio. With this estimation of energy consumption, we select the pair of wake-up intervals (w^h_{ij}, w^l_{ij}) that minimizes $E(w^h, w^l)$, where $(w^h, w^l) \in S_{ij}$.

Finally, the node chooses the minimum wake-up intervals w^h_i and w^l_i among w^h_{ij} and w^l_{ij}, respectively, i.e.,

$$w^h_i = \min\{w^h_{ij} \mid j \in N - \{i\}\} \tag{37}$$

and

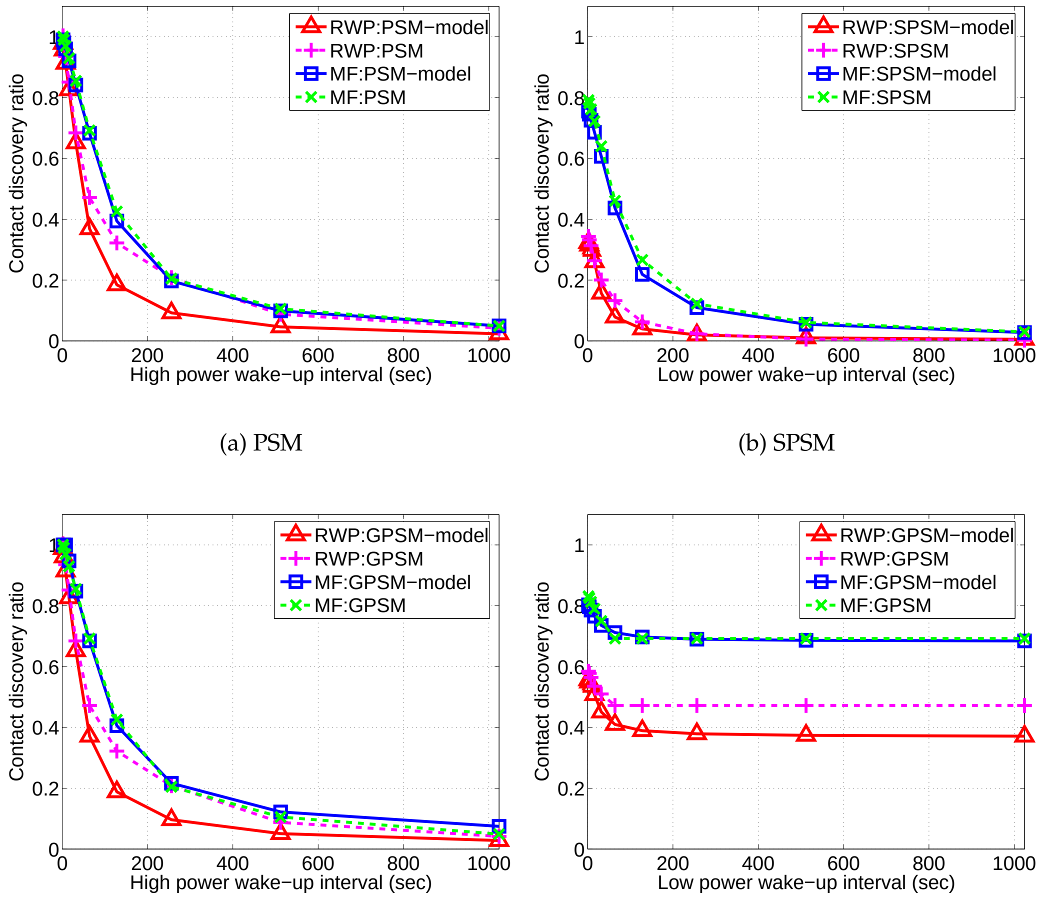
$$w^l_i = \min\{w^l_{ij} \mid j \in N - \{i\}\}, \tag{38}$$

where N is the set of nodes in the network. Then, the resulting (w^h_i, w^l_i) is the pair of wake-up intervals that consumes approximately the least amount of energy while satisfying the bandwidth constraints for all the links of node i.[2]

5.3 *Performance Evaluation*

Our goal in evaluating our traffic-aware optimization process is to show four things. First, we show that our analytical model can approximately predict the amount of discovered contact time for any particular wake-up intervals. If the predicted contact time is correct, then the wake-up intervals can be tuned to a particular traffic load. Second, we show that these wake-up mechanisms save significant energy in a DTN node and compare the use of

[2] If both w^h_i and w^l_i are infinite, we may want to set w^l_i to $\max W^l$ to discover unexpected contacts with minimum energy consumption.



(a) PSM

(b) SPSM

(c) GPSM(w^h)

(d) GPSM(w^l)

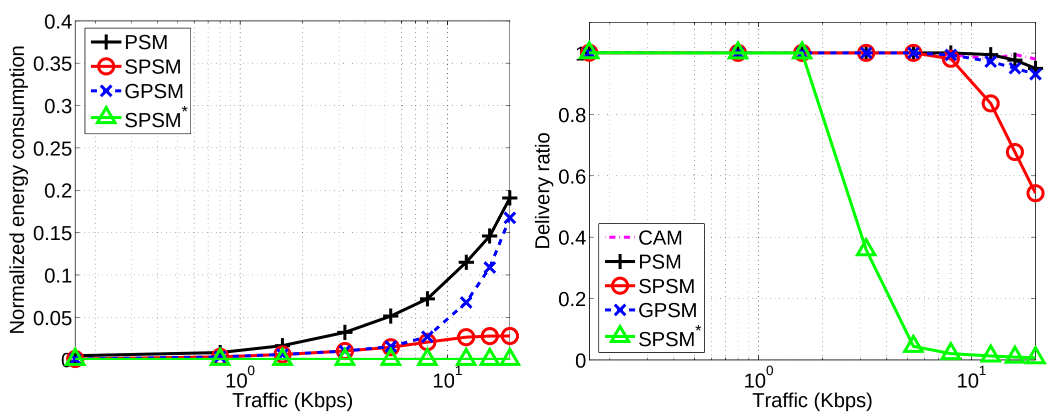
Figure 29: The actual contact discovery ratio compared to the estimated contact discovery ratio for both scenarios, MF and RWP, when wake-up intervals vary. Figure 29(c) is drawn when the low power wake-up interval w^l is 1024 seconds, and Figure 29(d) is drawn when the high power wake-up interval w^h is 64 seconds.

single and multiple radio search mechanisms. Third, we show that the energy efficiency of contact discovery varies under various node mobility scenarios and power management mechanisms. In all cases our generalized power management (GPSM) is flexible enough to adapt its behavior to mobility scenarios and performs as well as the other techniques. To show this we use the same RWP, MF, and UMassDieselNet mobility scenarios used in Section 4.4. Finally, we show that the usefulness of the low-power radio depends on the effective radio range of the high-power radio, in which high-power radios can achieve high throughput. As the effective radio range decreases, the additional low-power radio becomes more useful.

5.3.1 Contact Discovery

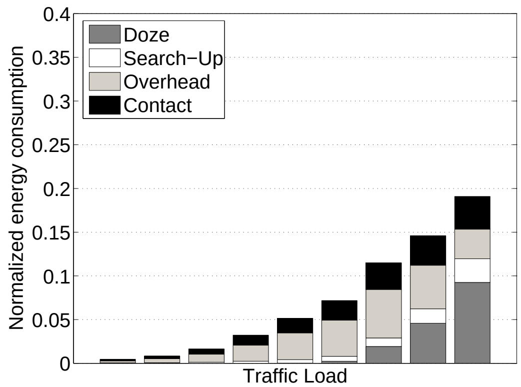
Figure 29 shows how close our estimation of contact discovery in Section 5.2 is in comparison to the actual contact discovery. The contact discovery ratio is the discovered contact time divided by the possible contact time that could have discovered by using the high-power radio in an always-on mode (CAM). Each line represents the estimated contact discovery ratio from our model and the simulated contact discovery ratio of each power management in RWP and MF as indicated. The UMassDieselNet scenario has the similar results, but not shown in the graph for visibility. Each scenario has a pair of nodes in a deployment area under the same simulation setting as in Section 4.4. In Section 5.2 we assume that contacts arrive according to a Poisson process and contact durations are constant. However, each mobility scenario may have different contact behavior. As a result, our estimation has error up to 14%, 5%, and 18% of the total contact time in RWP, MF, and UMassDieselNet, respectively. Nevertheless, Figure 29 shows that our analytical estimation approximately predicts the contact discovery ratio to the simulation results. These graphs are representative for other pairs of wake-up intervals in our evaluation.

5.3.2 Traffic-Aware Power Management

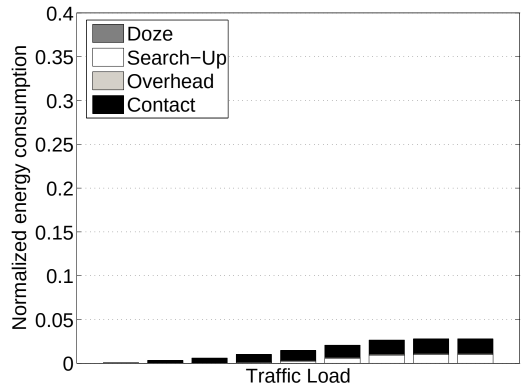


(a) energy consumption (b) delivery ratio

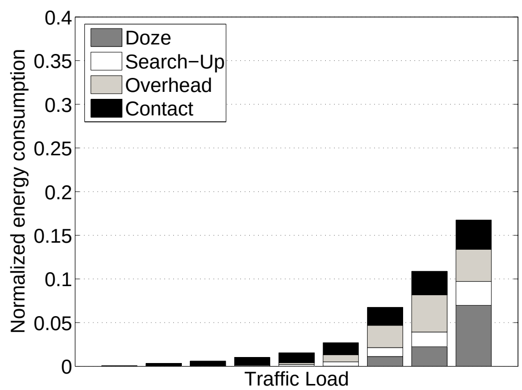
Figure 30: The impact of traffic-aware optimization to the power management in the RWP scenario



(a) PSM



(b) SPSM



(c) GPSM

Figure 31: The breakdown of energy consumption in the RWP scenario

Given the accuracy in determining the contact time resulting from a particular wake-up interval, we compare the schemes using our traffic-aware wake-up interval estimation. We consider the following metrics: (1) *normalized energy consumption* that is the total energy consumption of each power management case divided by that of CAM, and (2) *delivery ratio* that is the ratio of successfully delivered messages to (the total number of generated messages − the number of remaining messages in the network) by the end of a simulation. (3) *energy efficiency* that is discovered contact time divided by consumed energy. In simulations, a message loss occurs when a message buffer overflows at a node in the middle of a routing path because the node attempts to store messages beyond the limitation of its buffers.

To determine the routing paths of messages, we use a modified Dijkstra's algorithm that finds the shortest path to minimize the delivery delay of messages [30]. Since we assume to have statistical information about contacts, the minimum expected delay (MED) is used. The resulting routing paths are stored in the message headers to be used in source routing. Details can be found in [30].

For traffic load, each source generates 1KB messages according to a Poisson process with rate of one message per 1000, 200, 100, 50, 30, 10 and 8 seconds in RWP, one message per 32, 16, 8, 4, 2, 1, 0.8, and 0.75 seconds in MF, and one message per 32, 16, 8, 4, 2, 1, and 0.9 seconds in UMassDieselNet. To compare each scenario on equal footing, we selected these message generation rates to match the capacity of the responsive scenarios, and all yield PSM wake-up intervals between 2 and 1024 seconds. In RWP, each node selects a random destination. In MF, each node sends messages to the ferry. In UMassDieselNet, six random pairs of source and destination whose routing paths include the stationary node in the middle are selected. Each node has a limited buffer to store 16,000 messages per next hop node in RWP, 32000 in MF, and 60000 in UMassDieselNet, except that a ferry has an infinite buffer. These buffer sizes are selected to provide enough capacity to hold generated messages and received messages during the expected waiting time between discovered contacts with 90% probability of discovery. For traffic-aware power management, we assume that nodes exchanged the history of contacts and traffic load in advance and the network is stable [18, 30]. To provide the statistical knowledge of the network dynamics, we generate the movement of nodes for the whole simulation duration in advance and then use the same movement to run the simulations.

Figures 30 and 31 show the impact of traffic-aware optimization on the power management performance in the RWP scenario as the traffic load varies. We compare all of the schemes with CAM to observe how much energy can be saved. Also, we compare it with a SPSM when the traffic load information is not available. In such a case, we use a fixed wake-up interval that consumes the least amount of energy to discover a unit contact time in our simulation settings. This allows us to compare a traffic-aware mechanism with one that neglects any knowledge of traffic and focuses solely on energy efficient discovery. In our scenarios, SPSM with a wake-up interval of 1024 seconds is the most energy efficient mechanism for discovery and the corresponding curve is indicated by SPSM* in Figure 30.

Figure 30(a) shows that all mechanisms consume energy only from 1% to 18% compared to no power management. Also, it shows that as the traffic load increases, all mechanisms with traffic-aware optimization consume more energy because they choose smaller wake-up intervals to discover more contacts. Among these mechanisms, SPSM energy consumption is relatively flat. As the load grows, SPSM searches for contacts as aggressively as it can. However, SPSM relies on the low-power radio for contact discovery, which has insufficient range to discover enough contacts to handle its traffic. As a result, SPSM discovers less contacts, consuming less energy than other mechanisms as shown in Figure 30(a), and has a low delivery ratio when the traffic load is more than 5.4Kb/s as shown in Figure 30(b). On the other hand, PSM and GPSM achieve more than 90% delivery ratio for all traffic loads, which is close to the delivery ratio of CAM. Figure 30 also shows that GPSM delivers as many messages as PSM, but also saves more energy than PSM by utilizing the low-power radio for contact discovery. Finally, SPSM* consumes the least amount of energy for all traffic loads. However, it cannot discover enough contacts to deliver messages when the traffic load is heavy. Thus, the additional information about network load helps to enhance the network performance while saving energy.

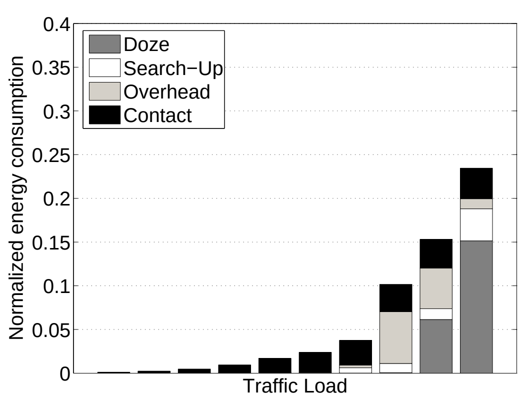
Figure 31 shows the breakdown of energy consumption for each power management scheme in the RWP scenario. The energy consumption is categorized into four classes: *doze*, *search-up*, *overhead*, and *contact*. The doze class corresponds the energy consumed by radios while dozing. The search-up class corresponds the energy consumed by radios while they are awake for searching. The overhead class corresponds the energy consumption by radios to transit between activities using different power such as turning on. Finally, the contact class corresponds the energy consumed by radios while having contacts.

Figures 31(a) and (c) show that radios consume relatively small amounts of energy for contacts while consuming significant amounts for discovering contacts for all traffic loads. Specifically, when the traffic load is moderate, the high-power radio is turned off for sleeping. Thus, the overhead to turn on the radio is relatively high for PSM and GPSM.

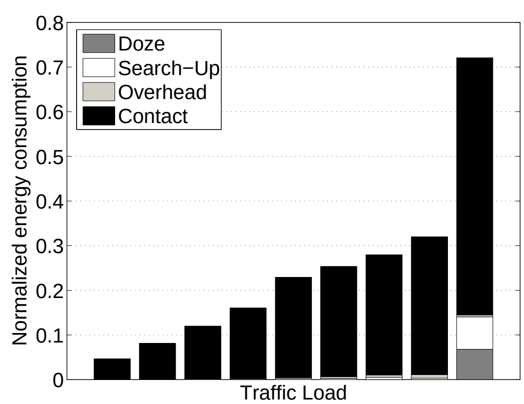
Also, when the traffic load is heavy, the radios search for contacts more often. Due to frequent wake-up's, some radios doze for sleeping, which increases energy consumption for dozing and decreases energy overhead to wake up the radios. Finally, the low-power radio used by SPSM consumes low enough energy that its energy overhead to wake up is negligible.

From this evaluation, we conclude the following. First, all of the techniques save considerable energy by sleep/wake-up cycling. Second, traffic-aware optimization saves more energy than pure optimization for discovery performance. Third, GPSM can choose appropriate wake-up intervals based on the expected traffic load and achieve the best performance for both energy savings and delivery rate.

5.3.3 Mobility Scenarios



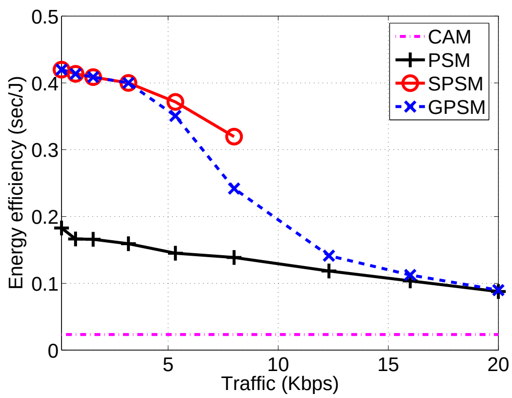
(a) MF (GPSM)



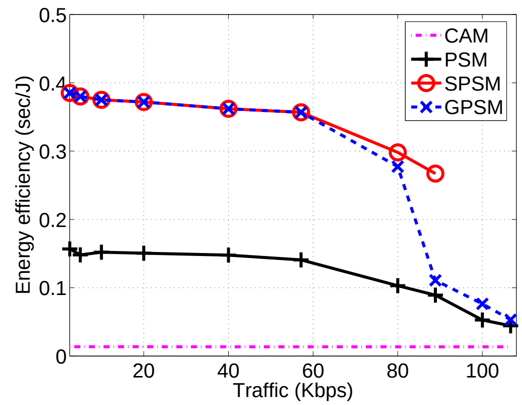
(b) UMassDieselNet (GPSM)

Figure 32: Energy consumption under various mobility scenarios

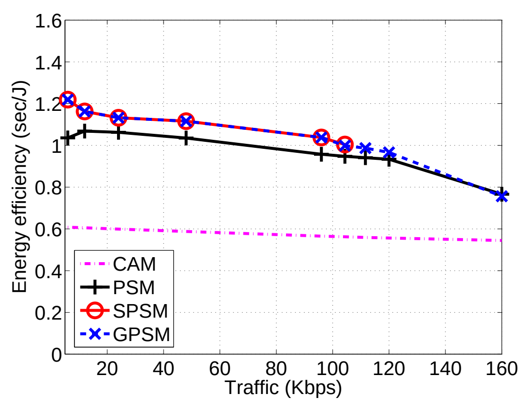
In this section, we show that the performance of power management varies depending on the node mobility scenarios. In Figure 32 the energy consumption of nodes using GPSM is broken down into four components as before in MF and UMassDieselNet scenarios. Figure 32(a) shows that in the MF scenario, when traffic load is light, nodes consumes negligible energy to search for contacts. When the traffic load is heavy, nodes use the high-power radio more aggressively to discover large amounts of contact time. Since a ferry takes relatively long time to revisit each node than to contact it, nodes spend more time on



(a) RWP



(b) MF



(c) UMassDieselNet

Figure 33: The energy efficiency of contact discovery under various mobility scenarios

searching, resulting in high energy consumption for search-up and sleep.

Figure 32(b) shows that in the UMassDieselNet scenario, the stationary node consumes most of its energy to contact other nodes. In the UMassDieselNet scenario, the stationary node is placed in a high-traffic area, so the inter-contact time is an order of magnitude less than that in other scenarios, resulting in low energy consumption for search-up/overhead and high energy consumption for contact. However, Figure 32(b) shows that our traffic-aware power management can save energy by discovering just enough contacts for the expected traffic loads. In our simulations, the delivery ratio of PSM and GPSM is close to that without power management.

In Figure 33, we show the energy efficiency of traffic-aware power management schemes in RWP, MF, and UMassDieselNet scenarios. We present the results only when the corresponding delivery ratio is greater than 90% to eliminate the case in which a scheme (i.e., SPSM) achieves the high energy efficiency at the cost of low delivery ratio. That is, the energy efficiency is assessed only for the case when most of the traffic is delivered.

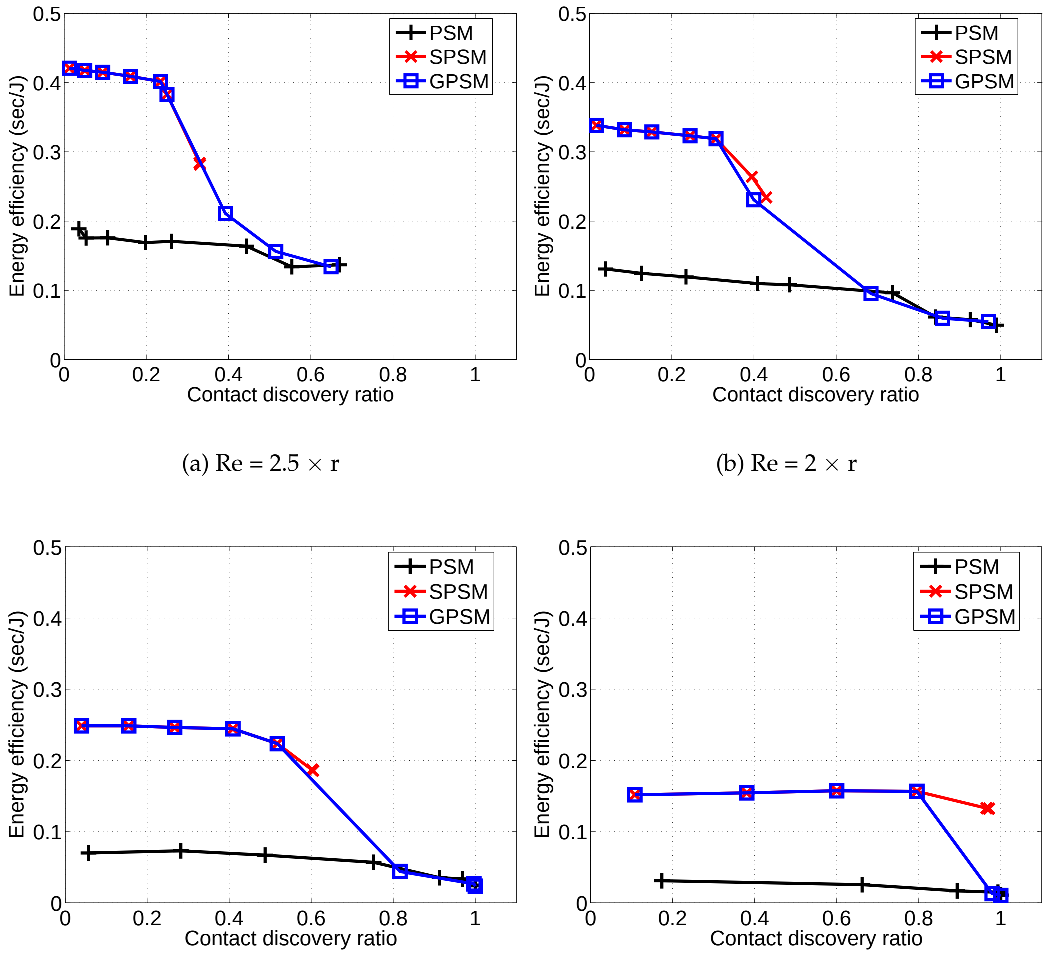
Figure 33(a) shows the results in the RWP scenario. When the traffic load is light, SPSM and GPSM achieve the highest energy efficiency because they can rely on the low-power radio with large wake-up intervals for contact discovery. When the traffic load is 5.4Kbps, GPSM starts to use both high-power and low-power radios to discover more contact time than can be discovered solely by the low-power radio, resulting in slightly lower energy efficiency than SPSM. SPSM does not discover enough contact time to deliver all the traffic loads, but it manages to deliver 90% of the traffic load up to 8Kbps and achieve the highest energy efficiency. In the meantime, GPSM and PSM discover enough contact time to deliver the traffic load up to 20Kbps.

Figure 33(b) shows similar results in the MF scenario. The only difference is that SPSM can discover contact time to deliver up to 89Kbps, which is relatively large ratio of traffic that can be delivered using CAM in the MF scenario. Because the ferry always enters the radio range of the low-power radio whenever it meets a node, the low-power radio can discover more contact time than in the RWP scenario.

Figure 33(c) also shows similar results in the UMassDieselNet scenario to those in the RWP scenario. The difference is that the energy efficiency of SPSM, PSM, and GPSM are close to one another compared to the other two scenarios. Because the stationary node spends most of the time on contact rather than on searching, the energy efficiency is similar for the same discovered contact time. When the traffic load is higher than 104Kbps, SPSM cannot discover enough contact time, delivering less than 90% of the traffic load.

5.3.4 Effective Radio Range

Discovering a contact does not mean that two nodes can exchange their messages with a high data rate immediately. When a signal strength is weak, messages may be lost and require several retransmissions before successful delivery. So, a typical 802.11 card selects



(a) Re = 2.5 × r

(b) Re = 2 × r

(c) Re = 1.5 × r

(d) Re = 1 × r

Figure 34: The impact of effective radio range of the high-power radio (*Re*) to the contact discovery performance in the RWP scenario, where *r* indicates the radio range of the low-power radio

its data rate based on the signal strength it receives, while the signal strength depends on many factors such as the distance between the nodes. Due to these facts, Ott et al. show that 802.11 cards have low throughput at the moment they discover each other [51]. When they become closer, the cards enter the time period in which they can achieve high throughput effectively. As the nodes move away from each other, their signal strength becomes weak and their throughput drops again. Therefore, a radio has a shorter range to perform transmission effectively than the range to discover other nodes. We refer this shorter radio range to an *effective radio range*.

In this section, we investigate the impact of the effective radio range on the performance of our power management mechanisms, especially, the usefulness of the low-power radio. That is, if the effective range of the high-power radio decreases to the low-power radio range, schemes using the low-power radio for discovery become more useful than the others because they can discover equivalent amount of contact for effective transmission while consuming less energy. The effective radio range can be a function of the distance between nodes. However, there are many other factors such as obstacles and transport layer protocols [51]. So, we abstract the model and compare four scenarios in which the effective radio range of the high-power radio is 1, 1.5, 2, and 2.5 times of the range of the low-power radio. Contact discovery of each mechanism is conducted as before. Meanwhile only the contact within the effective radio range of the high-power radio is considered as the discovered amount of the contact. To determine wake-up intervals, we use the same traffic load in Section 5.3.2. Each simulation was run for 100 days in simulation time.

Figure 34 shows that the discovery ratio of SPSM increases as the effective radio range decreases from 2.5 times to one times of the low-power radio range. GPSM also use the low-power radio more. In fact, when *Re* is 2.5 times of *r*, SPSM only discovers contact up to 35%. Meanwhile when *Re* is same to *r*, SPSM discovers contact more than 95%. Thus, the useful ness of the low-power radio increases as the effective radio range of the high-power radio approaches to the low-power radio range.

In summary, using the traffic load information, GPSM adaptively selects the wake-up intervals and achieves energy efficiency as high as SPSM when the traffic load is light. Also, it adaptively uses its high-power radio to achieve as good delivery ratio as PSM when the low-power radio could not discover enough contact to handle the expected traffic load. In addition, the usefulness of the low-power radio varies under the node mobility scenario as well as the effective range of the high-power radio. That is, the additional low-power radio becomes more useful as more contacts to achieve high throughput can be discovered by the low-power radio. Also, the energy efficiency of using the low-power radio increases as the sparseness of the network increases.

5.4 Summary

In this chapter we devise traffic-aware approximation algorithms for hierarchical power management mechanisms, in which nodes control two radio interfaces to discover contacts. In our algorithms, nodes control wake-up intervals of its radios to save energy while discovering enough contacts to deliver the traffic load in the network. Our simulation results from three mobility models show that our generalized power management mechanism balance energy efficiency and delivery performance by tuning the wake-up intervals of the two radios. In addition, when the traffic load can be estimated in advance, our approximation algorithms help nodes save significant energy while handling the expected traffic load. Finally, the relative energy efficiency of using the additional low-power radio increases as the nodes' searching time increases.

So far, we have developed power management mechanisms in DTNs by passively observing the characteristics of DTNs. However, we came to ask a more fundamental question: Is the DTN approach only a back-up approach to resolve networking issues when traditional mechanisms do not work? In the next chapter, we investigate how to use a DTN approach proactively for energy savings even when the traditional approaches work.

CHAPTER VI

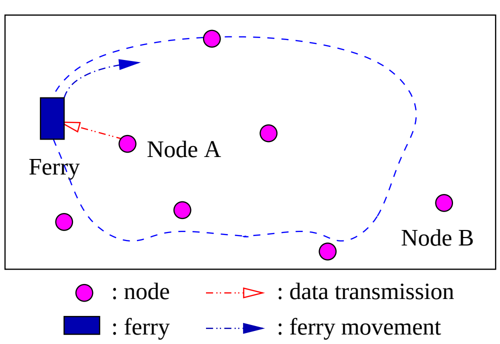
TRADING LATENCY FOR ENERGY USING MESSAGE FERRYING

6.1 *Introduction*

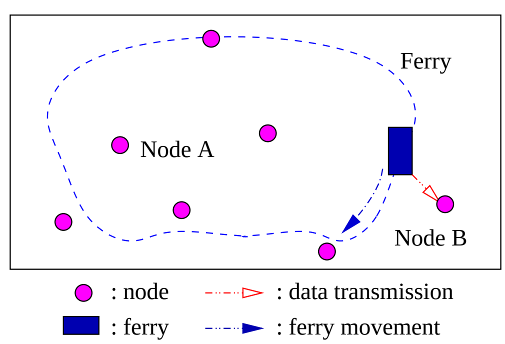
Mobile ad hoc networks (MANETs) consist of wireless nodes that relay data for one another to form a connected network. These networks provide rapid deployment and self configuration capabilities and have applications in a variety of environments such as battlefields, disaster recovery, and environmental monitoring. However, nodes in MANETs often have limited energy supplies. Therefore, efficient power management mechanisms are necessary to allow these networks to operate over a long period of time. In energy-limited devices, the wireless interface is one of the largest consumers of energy [45]. In addition to consuming energy during active communication, the wireless interface also consumes a significant amount of energy in the idle mode while listening for transmissions by other nodes. Studies have shown that energy consumption while listening to data is almost as high as that while actually receiving data [26, 70]. Thus, in the case of moderate traffic load, idle time is the dominating factor in energy consumption and nodes can save considerable energy by "sleeping," i.e., turning off or disabling their radios, if not communicating.

In sleeping nodes, data is stored until the nodes wake up. Such nodes can, therefore, achieve energy savings while trading off data delivery latency. For some applications, latency is not a critical issue. For example, when habitat monitoring nodes collect information periodically and send it to a central node, delivering data ten minutes later does not make much difference. Thus, for these "delay-tolerant applications," nodes can save more energy by sleeping longer, while increasing latency. For MANETs using a multi-hop routing approach [34, 53, 56, 57], energy saving techniques that end up trading off latency have been proposed in the literature [1, 20, 64, 65, 71, 74, 77, 82]. However, there are a number of unresolved problems in techniques that aim to achieve energy savings this way. First of all, sleeping nodes can cause a network to become disconnected, and as a consequence data cannot be delivered even if the network is densely deployed. Secondly, if nodes are mobile, the network topology might change during sleeping periods, making earlier routing information obsolete. Reconstructing routing tables or paths would consume additional energy. Finally, accumulating data for a long time and sending them out together increases contention in the network, which results in data loss or additional energy consumption due to retransmission.

In this chapter, we consider an alternative routing approach, *Message Ferrying (MF)* [78, 79]. In the MF approach, special nodes called *ferries* move around the area in which a network is deployed. These ferries are in charge of delivering messages among nodes as shown in Figure 35. When node A has a message for node B, it sends the message to a ferry when they are close to each other. Then, the ferry moves along its planned route. When the ferry becomes close to node B, it sends the message to node B. Similar routing approaches have been proposed for many applications, e.g., ZebraNet [35] to track wild life, DakNet [29] to provide high bandwidth Internet service in rural areas, and DataMule [32] to collect data from stationary sensors. These routing approaches have been developed mainly to deliver messages in sparsely deployed and partitioned networks.



(a) From node A to Ferry



(b) From Ferry to node B

Figure 35: An example of message delivery from node A to node B using MF

In this chapter, we consider the use of MF in a network with *densely deployed nodes*, and study how to achieve energy savings by trading off latency. The use of MF can increase data delivery delay over "traditional" MANET multiple routing protocols (e.g., DSR [34], AODV [56], and DSDV [57]). However, it has important features that enable the network to save energy compared to these multihop routing approaches. First, utilizing the knowledge of ferry location, nodes can sleep without degrading performance when ferries are out of communication range. Second, ferries are in charge of data delivery, so nodes do not need to wake up to form a connected network because the ferry mobility eventually connects the network. Also, topology changes in MF do not require any overhead to reconstruct routing tables. Finally, the movement of ferries allows nodes to transmit data at different times according to their locations and decreases contention among nodes.

To exploit these features of MF, we propose a power management framework for both stationary and mobile nodes. In our framework, nodes switch among different power management modes according to the knowledge of ferry location. We evaluate our schemes using ns-2 simulations and compare them with Dynamic Source Routing (DSR) [34] with and without an idealized power management scheme. Our simulation results show that MF can achieve significant energy savings. In contrast, power-managed DSR reduces energy consumption at the price of significantly lower delivery rate. For example, MF achieves energy savings up to 95% compared with DSR without power management and delivers over 98% of messages under all traffic loads. However, power-managed DSR delivers as low as 20% of the messages to achieve similar energy savings. In addition, MF shows robust performance for mobile nodes, while the performance of DSR suffers significantly.

The remainder of this chapter is structured as follows. In section 6.2, we describe the network model used in our study. Section 6.3 presents the power management mechanisms for MF and Section 6.4 shows our simulation results. We conclude the chapter in Section 6.5.

Table 8: Power usage parameter values used in the simulation (unit: W)

Activity	Transmit	Receive	Idle	Doze	Off
Power	0.2818	0.2053	0.1791	0.0141	0

6.2 *Network Model*

We consider networks consisting of stationary or mobile nodes in a deployment area. Nodes communicate with each other via wireless interfaces. We assume that nodes are identical and are limited in resources. That is, nodes are equipped with the same radios and have the same buffer size and energy supply. In addition, nodes have knowledge of their location and time, e.g., through global positioning system (GPS) or other localization mechanisms.

6.2.1 Energy Consumption

In this chapter, we consider only communication energy consumption and do not account for energy consumption of other sources such as computation or mobility. The energy consumption of a wireless interface depends on its activities, i.e., *transmitting*, *receiving*, *idling* (when listening to the wireless medium without transmitting nor receiving), *dozing* (when the wireless interface is inactive), and being *off* (when the wireless interface is turned off and consumes no energy.) The amount of energy consumption in each activity is assumed based on the studies in [26]. When dozing, a node consumes an order of magnitude less energy than when idling, while an idling node consumes energy at the same order of magnitude as a receiving or transmitting node. Table 8 shows the power usage parameter values used in our simulations. In addition, we consider the transition overhead to turn on the radio, from being off to idling, because it consumes considerable energy.

6.2.2 Message Delivery

We consider two approaches for data delivery in the networks, namely *multihop routing* and *message ferrying (MF)*. In the multihop routing approach, nodes relay messages for one another such that messages can be forwarded from the source to the destination via intermediate nodes. In the MF approach [78], special nodes, called *ferries* move around the deployment area and are responsible for delivering messages for nodes. By carrying messages from the source to the destination, ferries are able to provide communication service to nodes.

In the MF scenario, we consider a network consisting of multiple nodes and a single ferry.[1] We assume that the ferry has ample resources such as large storage and sufficient power supply.[2] To initiate message exchange with nodes, the ferry broadcasts Hello messages, called *beacons*, periodically and nodes in the radio range of the ferry respond to the ferry if they desire to exchange messages. Thus, nodes do not need to form a connected network. Instead, they are required to detect ferry arrival in their neighborhood by listening for beacons and then to exchange messages with the ferry. Figure 36 shows an example of message delivery triggered by beacons in the MF. In this example, the ferry moves down along a central line, while nodes A, B, and C are located beside the line. Initially, the ferry broadcasts a beacon, which is received by nodes A and C. Because it has messages to send to node B, node A sends a response to the ferry, followed by messages. Meanwhile, node C ignores the beacon since it does not have messages to send nor to receive. As it moves down, the ferry keeps broadcasting beacons. Then, node B hears the third beacon. Because the beacon indicates pending messages for itself, node B sends a response to the ferry. After receiving the response, the ferry sends stored messages to node B.

To specify the movement scenarios, we assume that the ferry is an existing entity whose movement is assumed not controllable for the purpose of assisting communication and is required for other purposes.[3] To investigate ideal and practical movement of the ferry, we assume that the ferry moves on a fixed route with either a *strict* schedule or a *loose* schedule

[1]When a network has multiple ferries, more issues arise, such as dealing with coordination and different movement patterns of ferries [80]. Thus, we leave the consideration of multiple ferries to future work.

[2]In the multihop routing scenario, it doesn't help that much for a network to have such a node because the other nodes consume as much energy as before regardless of the additional resources and limit the network lifetime.

[3]As an example of the MF approach, a shuttle bus in a national or amusement park can be used as a ferry to collect information from sensors deployed in the park. Since the movement of the shuttle bus is needed for transportation, the energy for the ferry movement does not need to be considered. Also, the energy for the communication part of the ferry is easily rechargeable in such a vehicle, so it does not limit the lifetime of the network.

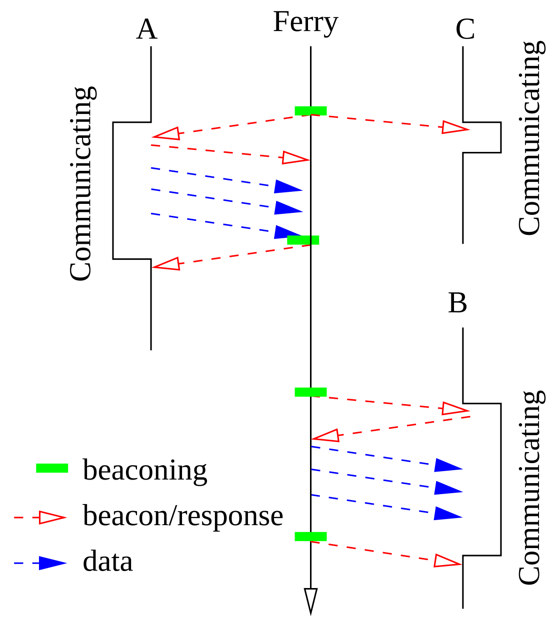


Figure 36: Message delivery triggered by beacons in Message Ferrying

in which nodes know the route. With a strict schedule, the ferry arrives at each location as it is scheduled. Thus, nodes can estimate when to meet the ferry precisely. With a loose schedule, the ferry is allowed to slow down or pause, which makes it hard to predict the ferry arrival at each location.

6.3 *Power Management In Message Ferrying*

6.3.1 Power Management Framework

In this section, we describe the framework of our adaptive power saving mechanism. In the mechanism, a node is in one of three power management modes: *sleeping*, *searching*, and *communicating*. In the sleeping mode, a node sleeps (i.e., dozes or turns off its radio) because the ferry is out of the communication range. In the searching mode, a node periodically wakes up to listen for a beacon because of insufficient information about ferry movement. Finally, in the communicating mode, a node wakes up frequently to communicate with the ferry in its radio range. To describe the wake-up behavior of a node in each mode, we define three time periods: *wake-up interval*, *beacon period*, and *active window*. A wake-up interval is the time between consecutive wake-up events at a node. A beacon period is the time between consecutive beacon generations by the ferry. Finally, an active window is a fraction of a beacon period, starting from the beginning of a wake-up interval.

These periods are used in the searching or communicating modes as follows. A node wakes up every wake-up interval, which is a multiple of a beacon period. If it does not receive a beacon within an active window, it goes to sleep until the beginning of the next wake-up interval. When a node receives a beacon, if it has any messages to send or to receive, it stays awake for a beacon period. Otherwise, it goes to sleep again until the beginning of the next wake-up interval.

Transitions among the power management modes are triggered by timers or beacon receptions and are shown in Figure 37. Initially, a node estimates the shortest time after which it can communicate with the ferry, called *sleeping time*. Then, it enters the sleeping mode and sets a timer to expire after the sleeping time. When the timer expires, the node estimates its sleeping time, if needed. If it is positive, the node remains in the sleeping mode. Otherwise, the node switches to the searching mode to listen for a beacon. After receiving the first beacon, it switches to the communicating mode. Finally, if the node does not receive a given number of beacons consecutively, it switches to the sleeping mode.

Depending on the movement scenarios, the transition among the modes could be optimized. For example, in the case that the ferry moves on a strict schedule and nodes are stationary, a node can estimate the exact time to communicate with the ferry. Thus, the node may alternate only between the sleeping and communicating modes based on its estimation, without passing through the searching mode. In the case that the ferry moves on a loose schedule and nodes are mobile, at the ferry departure, a node may switch from the communicating mode to the searching mode, instead of the sleeping mode, if the node resides within the range in which it may meet the ferry again. In addition, the node in the searching mode may switch to the sleeping mode, if it moves away from the range in which it may meet the ferry again.

Figure 38 shows an example scenario in which a node switches its power management mode according to the location of the ferry. A node is in the sleeping mode when the ferry is out of radio range. When it expects to meet the ferry in the near future, it switches to the searching mode and wakes up periodically to listen for a beacon. After receiving the first

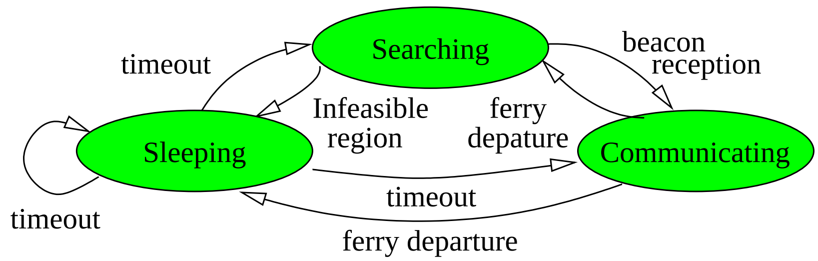


Figure 37: Transition among power management modes

beacon, it switches to the communicating mode and frequently wakes up to communicate with the ferry. Finally, when the ferry leaves the radio range, the node switches to the sleeping mode again.

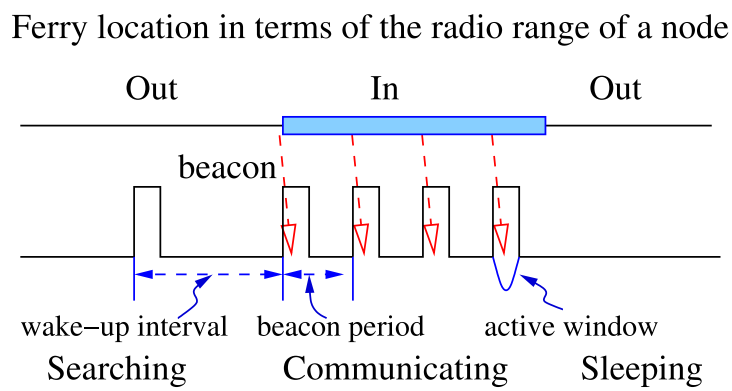


Figure 38: Power management modes of a node depending on the ferry location

The power management at each mode is designed to save energy based on the characteristics of each mode. Specially, when a node sleeps, it decides whether to doze or to turn off its radio based on the duration of sleeping. If the energy consumption of dozing for the duration is greater than the transition overhead to turn on the radio, a node turns off its radio. Otherwise, it dozes. In the sleeping mode, sleeping time is often long because of the physical movement of the ferry. So, a node turns off its radio. The estimation mechanisms of the sleeping time will be described in Sections 6.3.2 and 6.3.3. In the searching mode, a node periodically wakes up to listen for a beacon and sleeps if it does not receive a beacon within an active window. The setting of this wake-up interval reflects the trade-off between energy savings and the delivery delay of messages. A longer wake-up interval conserves more energy, but may result in missing beacons that leads to longer delay. Finally, in the communicating mode, a node communicates with the ferry, which is within its radio range. That is, a node wakes up every beacon period to see if it needs to exchange messages with the ferry. In this way, when the ferry receives messages from other nodes destined to this node during the time, the messages can be delivered quickly.

6.3.2 Estimation of Sleeping Time for Stationary Nodes

In this section, we explain how to estimate the sleeping time of stationary nodes as well as mobile nodes. To assist the explanation, we use the following notations.

A ferry location is represented as $F(t)$ at time t. The route itself is defined as F where $F = \{F(t)|\ t \geq 0\}$. For a loosely scheduled scenario, $F(t)$ represents the estimated ferry location at t, assuming the ferry moves at its maximum speed without pause. Similarly, a node location at time t is denoted as $N(t)$. If a node is stationary, the location is denoted as N. The maximum speed of the ferry and nodes are v_F and v_N, respectively. A beacon period is p and the radio radius of nodes and the ferry is r. Finally, the current time is t_0. Here, we assume t_0 as a multiple of p without loss of generality.

6.3.2.1 Strictly Scheduled Ferry Movement

When the ferry moves on a strictly scheduled route, a stationary node can estimate its sleeping time easily by finding when the ferry arrives and leaves its radio range. Figure 39 shows how sleeping time is calculated. Initially, the ferry is at the location of $F(t_0)$ at t_0, which is in the radio range of a node. The ferry leaves the radio range at t_1 and enters again at t_2. A node communicates with the ferry until t_1, when the ferry departs its radio range. Then, the node sleeps until t_2, when the ferry enters its radio range again. Since a node only needs to wake up in the beginning of a beacon period, the sleeping time estimation needs to be adjusted to reflect the ferry arrival as a discrete time event with time granularity of a beacon period p, starting from time zero.

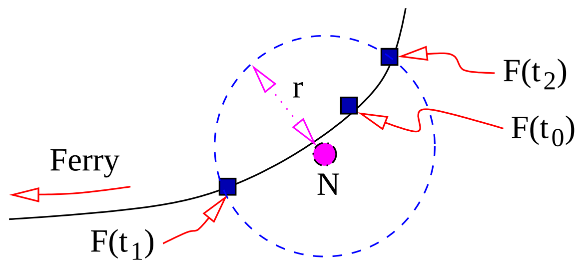


Figure 39: The intersection of a ferry route and the radio range of a node

6.3.2.2 Loosely Scheduled Ferry Movement

This scenario is an extension of the previous scenario. A stationary node estimates the minimum amount of time that the ferry takes to enter the next intersection between the ferry route and the radio range of itself. In fact, the ferry may take longer to enter the intersection because it may slow down or pause in the middle. Thus, a node assumes that the ferry moves at its maximum speed and sleeps only for the minimum amount of time that the ferry takes to enter the next intersection. After sleeping, a node switches to the searching mode and wakes up periodically to listen for a beacon.

6.3.3 Estimation of Sleeping Time for Mobile Nodes

In this section, we explain how to estimate the sleeping time of mobile nodes assuming no knowledge of the future movement of the nodes.[4]

6.3.3.1 Strictly Scheduled Ferry Movement

In this scenario, mobile nodes utilize the precise schedule of the ferry to estimate their sleeping time. When estimating its sleeping time, a node finds the earliest possible time that it can meet the ferry. To calculate this time, a node assumes that it will move directly toward the future location of the ferry at its maximum speed v_N. At time t, if the distance between the future locations of the ferry and the node is greater than r, it is not feasible for the node to be in the radio range of the ferry at t. Thus, the earliest possible time for a node to meet the ferry is the earliest time when the distance between the future locations

[4]If a node knows its future movement $N(t)$, the sleeping time can be estimated as if the node were stationary, on the origin of the coordinate, while the ferry moves on $F(t) - N(t)$.

of the ferry and the node becomes less than r. That is, when time is incremented by p, if there exists a minimum non-negative integer k that satisfies

$$|F(t_0 + k \cdot p) - N(t_0)| - v_N \cdot k \cdot p < r, \tag{39}$$

the node will not meet the ferry for a period of $(k-1)p$. Thus, the node can sleep for $(k-1)p$. After sleeping, the node determines k from Equation 39 again based on its current location. If k is greater than one, it sleeps again. If k is less than or equal to one, the node switches to the searching mode. In the searching mode, a node calculates k periodically to check whether it has left the radio range of the ferry so that it avoids waiting for a long time in case of losing beacons. If it has departed the radio range, it switches back to the sleeping mode. Otherwise, it stays in the searching mode.

Figure 40 illustrates the above procedure. Currently, a node is located at $N(t_0)$. As time is incremented by p, the future locations of the ferry are as follows: $F(t_0 + p)$, $F(t_0 + 2p)$, and so on. Assuming the node moves toward the future location of the ferry, the distance between the future locations is the distance between $F(t)$ and the tip of an arrow, where the length of the arrow represents the distance that a node can move at its maximum speed. Therefore, if the tip of the arrow lies outside of the radio range of the ferry, the node cannot enter the radio range of the ferry by that time. Thus, a node finds the earliest time for the tip of the arrow to enter the radio range of the ferry at its future location and sleeps until right before that time.

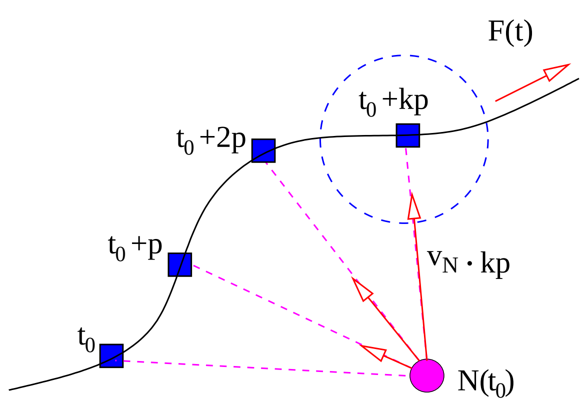


Figure 40: Sleeping time estimation when a node movement is not known in advance

6.3.3.2 Loosely Scheduled Ferry Movement

In this scenario, a node cannot easily estimate when it will encounter the ferry. However, it can estimate when it has no chance of encountering the ferry. Clearly, if a node is far away from any location of the ferry route, the node cannot communicate with the ferry. To formulate the problem, denote the distance between a node location $N(t_0)$ and the ferry route F as $d(F, N(t_0)) = \min_t |F(t) - N(t_0)|$. The *feasibility* of a node receiving a beacon is defined as follows:

Definition 1 *A node is in the* feasible *radio range of F if the distance between its current location,* $N(t_0)$*, and the ferry route F is less than a given radio radius r: that is, if*

$$d(F, N(t_0)) = \min_t |F(t) - N(t_0)| < r. \tag{40}$$

If a node is in the feasible radio range, it may receive a beacon. Otherwise, it will not receive any beacon. Therefore, estimating the sleeping time of a node is equivalent to finding the earliest possible time that the node enters the feasible radio range of F.

To estimate the sleeping time, a node assumes that it moves directly toward the closest location of the ferry route at its maximum speed, v_N. Then, the earliest possible time for a node to enter the feasible radio range of F is the earliest time when the distance between the ferry route and the future location of the node becomes less than r. In other words, if there is the minimum positive integer k such as

$$d(F - N(t_0)) - v_N \cdot k \cdot p < r, \tag{41}$$

the node will take at least kp to enter the feasible radio range of F. Thus, the node can sleep for $(k-1)p$. Here k is obtained from the following equation:

$$k = \lceil \frac{d(F - N(t_0)) - r}{v_N \cdot p} \rceil. \tag{42}$$

If k is equal to or less than one, a node switches to the searching mode with a default timeout value. At the timeout, it checks whether it leaves the feasible area using Equation 41.

Figure 41 illustrates an example. A node is currently located at $N(t_0)$. The closest location of the ferry route to $N(t_0)$ is the location where its tangential line and the line connecting the location and a node location intersects at 90 degree. Assuming the node moves toward the intersection, the tip of an arrow is the future location of a node at time kp after moving at its maximum speed v_N. If the tip of the arrow lies outside of the radio range of the intersection at time t, the node cannot enter the feasible radio range by that time. Thus, a node finds the earliest time for the tip of the arrow to enter the radio range from the intersection and sleeps until right before the time. When a node enters the feasible area, it switches to the searching mode. If a node rarely encounter the ferry, the node can save energy by increasing its wake-up interval without missing the ferry in its radio range most of the time. However, long wake-up interval also decreases the probability that a node will detect the ferry in its radio range. In the next section, we show the trade-off between energy savings and delivery delay by varying wake-up intervals.

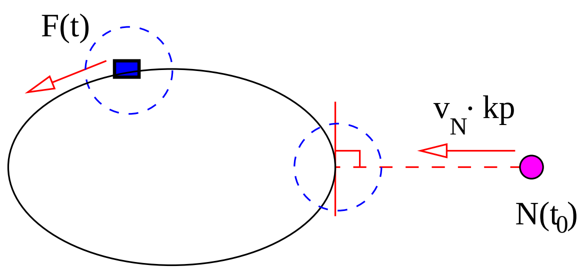


Figure 41: Checking whether a node is in the feasible radio range of a ferry route

6.4 *Performance Evaluation*

In this section, we demonstrate the trade-off between energy consumption and latency provided by the MF power-management scheme described in Section 6.3. To that end, we use simulations to compare the energy consumption and latency performance of a MANET deploying the MF scheme with one using multihop routing based on the use of Dynamic Source Routing (DSR) [34]. We choose Dynamic Source Routing (DSR) because it was determined to be the most efficient multihop routing protocol in [17].[5] In order to provide a

[5]Other routing protocols, not compared in [17] but used in the design of power management mechanisms, tend to have specific movement or location restrictions. Thus, they may cause more overhead than DSR if

fair comparison we consider using DSR in MANETs along with an ideal power management scheme that, while not realizable, provides a bound on the best possible performance of such networks.

The choice of proper power management depends on network topology and the capability of nodes [1, 20, 64, 65, 71, 74, 82]. Assuming minimal spatial redundancy and no secondary low-power channel, synchronous and asynchronous wake-up mechanisms are the basic wake-up approaches to use.[6] Between them, the asynchronous approach is considered to consume more energy than the synchronous approach because nodes usually have to stay awake for a longer time to overlap their awake time with those of their neighbors [82]. Because we are only interested in bounding the performance of MANETs using DSR, we use an idealized synchronous power management scheme. We define three time periods: *wake-up interval*, *awake period*, and *active window*. The wake-up interval is the time between consecutive wake-up events at a node. The awake period is similar to a beacon period in MF and is the time unit for a node to stay awake for message exchange. The active window is a fraction of an awake period, starting from the beginning of a wake-up interval. A node wakes up at the start of a wake-up interval and sends out data or route probing messages, if any. If a node sends or receives any messages within an active window, it stays awake for an awake period to participate in the upcoming communication. Otherwise, it sleeps until the beginning of the next wake-up interval. If it receives any messages during an awake period, it stays awake for another awake period.

6.4.1 Simulation Methodology

We use ns-2 simulations to compare the performance of MF and DSR with power management [2]. We also compare them with DSR without power management, called *Continuous Aware Mode (CAM)*. We consider the following four metrics:

- *energy consumption per node*: an average energy consumption per node in the network

used in general environments: For example, geographic forwarding requires frequent broadcasting, while not improving energy savings if the network density is low [20].

[6] The power management of MF can also be extended to utilize the spatial redundancy or secondary low-power channel, if they exist. In this chapter, we consider basic wake-up mechanisms only as an initial step.

- *delivery delay*: an average delay per delivered message
- *delivery rate*: the ratio of successfully delivered messages to the total number of generated messages
- *energy cost*: the average energy consumption to deliver a unit message, which is the total energy consumption divided by the number of delivered messages.

In simulations, we use the following default parameters, unless specified otherwise. Our network topology consists of 50 nodes, which are randomly located in a 2000m $\times$ 500m region. We use 802.11 MAC and the default power setting of ns-2. For example, the radio range is 250m and the data transmission rate is 2Mb/s. Additional power usage parameters are shown in Table 8. The energy used for node mobility is not counted because we assume nodes have other means to cause their movement. For example, they could be devices carried by people or sensors attached to animals, where their movement occurs due to the physical movement of people or animals. In MF, the ferry uses the same setting as nodes. Also, the energy used for ferry mobility is not counted because we assume the mobility already exists for other purposes.

To generate traffic, 30 pairs of source and destination nodes are randomly selected and each source chooses a random start time between 10 and 500 seconds. The sources send out messages at a constant rate of one message every 30 seconds for 3000 seconds. We define the traffic load as the total number of bytes generated from all sources in the entire simulation. Each message is of size 1KB and has a timeout value of 5000 seconds after which messages not reaching their destinations are discarded. Each node has a buffer to store 700 messages, while the ferry has an unlimited buffer space. Each simulation runs for $10,000$ seconds and each data point is the average of five runs.

In the implementation of power management, the beacon period in MF and the awake period in DSR are set to 2 seconds and the active window is 500ms. In addition, we use 10 beacon periods as a wake-up interval in loosely scheduled ferry movement scenarios. Finally, we simulate the energy consumption for a node to turn on its wireless interface. While the amount of consumption depends on devices, the time to resume the radio was measured as 100ms for three wireless interfaces in [70]. To assign reasonably large energy consumption, we use 0.05636J as the transition overhead to turn on the radio, which is equivalent to the amount of energy to transmit data for 200ms.[7] With this overhead, a node turns off its radio only if the expected sleep time is great enough to save energy beyond the overhead to turn on the radio back in both MF and DSR approaches. Otherwise, the node dozes and consumes no additional energy to enable the radio back.

To simulate node movement scenarios, we use the Random Way-point model [17] as follows. Each node selects a random destination in the region and moves toward the destination at a speed selected randomly between 0 and 10m/s. When it reaches the destination, it pauses for a *pause time*, which is exponentially distributed with an average of 10 seconds. When the pause time is up, nodes select another random destination to move toward. The ferry moves along a rectangular route, which has $(100, 100)$ and $(1900, 400)$ as two vertexes on its diagonal. As a result, the radio range of the ferry swipes through the whole simulation area as the ferry moves along its route. In the scenario of strictly scheduled ferry movement, the ferry moves at a constant speed of 10m/s. In the scenario of loosely scheduled ferry movement, the ferry moves at a constant speed of 10m/s on the edges of the route and pauses at four vertexes for a pause time, which is exponentially distributed with an average of 50 seconds.

6.4.2 Impact of Traffic Load

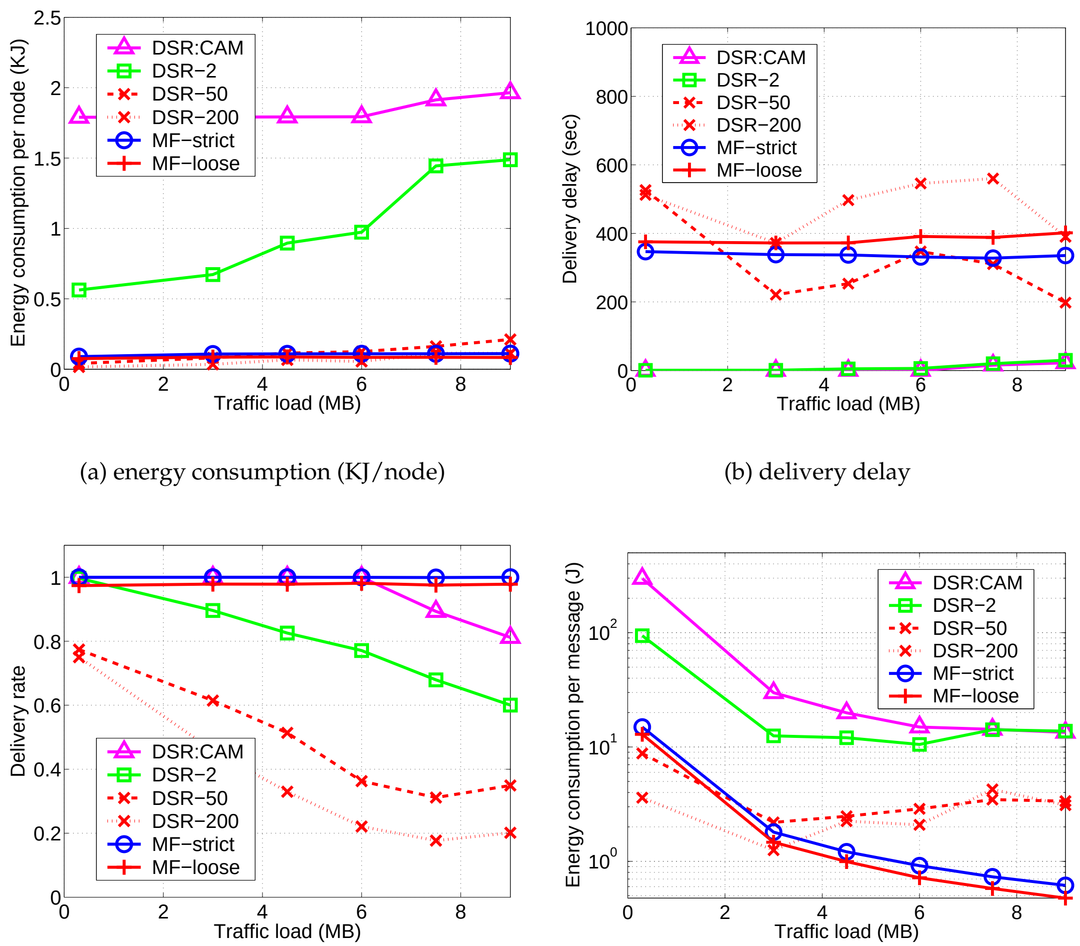
In this section, we evaluate the performance of MF and DSR under different traffic loads to show the relative robustness of the MF approach while the DSR approach suffers as more traffic load is injected to the network.

6.4.2.1 Stationary Nodes

We first compare the performance of MF and DSR when nodes are stationary. To vary the traffic load, we use message generation intervals of $300, 30, 20, 15, 12$, and 10 seconds.

[7] We simulated various overhead from the amount of energy for a node to transmit data for 50ms to that for 1 second. As the overhead increases, the energy consumption increases. However, the amount of increase was too small to make any difference in our comparison.

In Figure 42, we use DSR-x to represent the case of DSR with power management whose wake-up interval is x seconds, where x is $2, 50$, and 200 seconds. DSR:CAM represents the case of DSR without power management. We also use MF-strict to represent the case of MF with power management where the ferry moves on a strict schedule and MF-loose to represent the case where the ferry moves on a loose schedule.



(a) energy consumption (KJ/node)

(b) delivery delay

(c) delivery rate

(d) energy cost (J/message)

Figure 42: The impact of traffic loads when nodes are stationary

Figure 42(a) shows the average energy consumption of nodes. Here, MF and DSR with large wake-up intervals (e.g., 50 or 200 seconds) significantly outperform DSR:CAM and DSR-2 under all traffic load. For example, in case of the traffic load of 6MB, MF and DSR-200 consume only 0.2J, while DSR-2 and DSR:CAM consume five times or nine times of that, respectively. This is expected because nodes sleep for longer time.

Figure 42(a) also shows that increasing traffic load affects the energy consumption of DSR more than that of MF. In DSR, increasing the number of messages increases the number of transmission multiple times because of relaying the messages. Also, when the power management is used, nodes require to stay awake more to forward more messages. In fact, DSR-2 increases energy consumption faster than DSR:CAM as the traffic load increases, which shows that increasing idle time consumes more energy than increasing transmission by itself in DSR-2.

In Figure 42(b), we show the average delivery delay of messages. The delivery delay of MF is high because of the physical movement of the ferry. In the simulations, the ferry takes at least 420 seconds to come back to the same location. In DSR, using large wake-up interval also increases delivery delay because nodes store messages until the next wake-up interval before relaying if they are asleep. As a result, the delivery delay of MF, DSR-50, and DSR-200 lies in the range of 200 to 600 seconds, while that of DSR:CAM and DSR-2 is under 20 seconds for all traffic load.

Figure 42(c) shows the delivery rate. MF delivers most of the messages regardless of ferry movement scenarios. Meanwhile, DSR delivers fewer messages as the wake-up interval increases because nodes accumulate more messages and send them out at the same time, which increases contention. As the contention level increases, more messages are dropped. Similarly, as traffic load increases, DSR delivers fewer messages due to contention. For example, when the traffic load is 9MB, DSR delivers only 80, 60, 35, and even 20% of messages if CAM or power management with 2, 50, or 200 wake-up interval is used, respectively. However, MF delivers 98% of the messages under all traffic load.

In addition, Figure 42(c) shows that MF delivers less in a loose schedule scenario than in a strict schedule scenario because of infrequent wake-up in the searching mode. In a loose schedule scenario, a node wakes up only once every ten beacon period in the searching mode. So, it may miss a ferry, which passes through its radio range. If a node keeps missing the ferry and stores messages more than 5000 seconds, the messages are dropped. However, the loss rate is only 2%.

Figure 42(d) shows the energy cost on a log scale. As the traffic load increases, the energy cost of MF and DSR:CAM decreases because more messages are delivered without increasing energy consumption significantly. In DSR with power management, when the traffic load is low, energy consumption due to periodic wake-up dominates the total energy consumption. So, energy cost per message decreases as the load increases. When the traffic load is high, more messages are lost due to contention, leading to high energy cost.

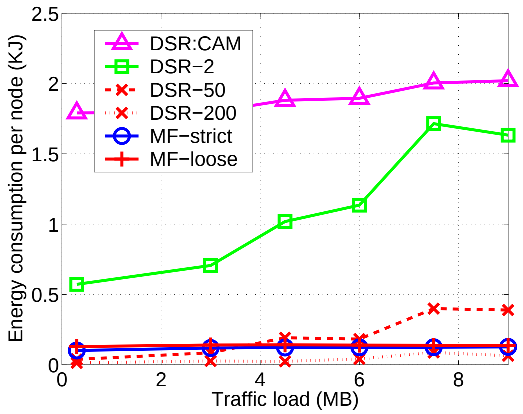
Finally, Figure 42(d) shows that MF has less energy cost in a loose schedule scenario than in a strict schedule scenario because of less energy consumption. In the loose schedule scenario, a node spends more time in the searching mode and less time in the communicating mode out of the total simulation time than in the strict schedule scenario. Since a node wakes up ten times more often in the communicating mode than in the searching mode, the total energy consumption of a node in the loose schedule scenario is less than that of a node in the strict schedule scenario. As a result, the energy cost may become smaller in the loose schedule scenario than in the strict schedule scenario.

6.4.2.2 Mobile Nodes

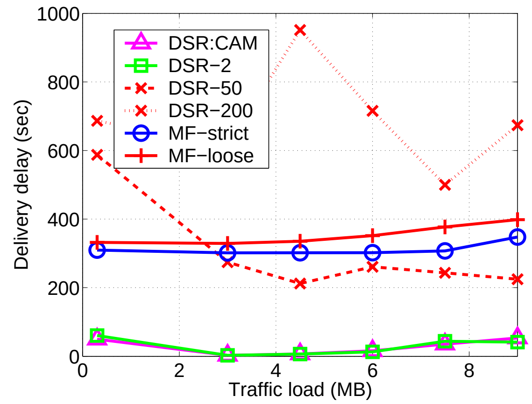
In this section, we evaluate the performance of MF and DSR when nodes are mobile. The results shown in Figure 43(a) and (d) are similar to those of the stationary node case. We, therefore, focus on the different features that show up in the simulation experiments.

Figure 43(b) shows the delivery delay. Compared with the stationary node case, MF delivers faster when nodes are mobile because mobile nodes meet the ferry more often. On the other hand, DSR delivers slower when nodes are mobile. Since the node mobility changes topology, nodes are required to probe routing paths before sending out messages if the change occurs. This waiting time for the route probing accounts for the increase in delivery delay.

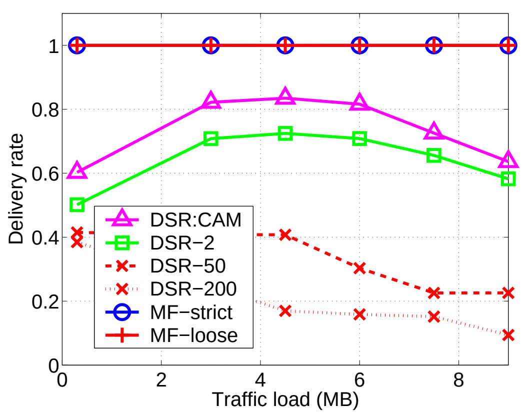
In Figure 43(c), we show the delivery rate. While MF delivers most of the messages under all traffic loads, the delivery rate of DSR varies significantly. In DSR, a node detects a route change by the failure of message transmission. Thus, it always loses the first message after a route change. In fact, DSR:CAM and DSR-2 have lower delivery rate when traffic



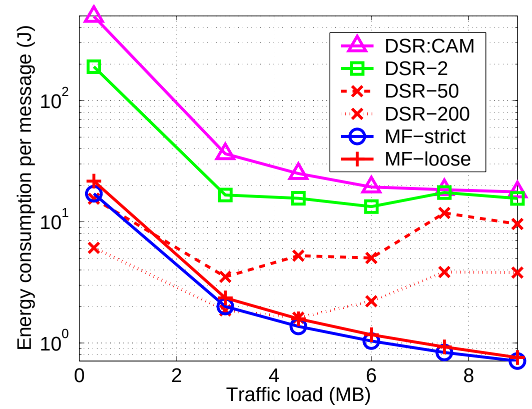
(a) energy consumption (KJ/node)



(b) delivery delay



(c) delivery rate



(d) energy cost (J/message)

Figure 43: The impact of traffic loads when nodes are mobile

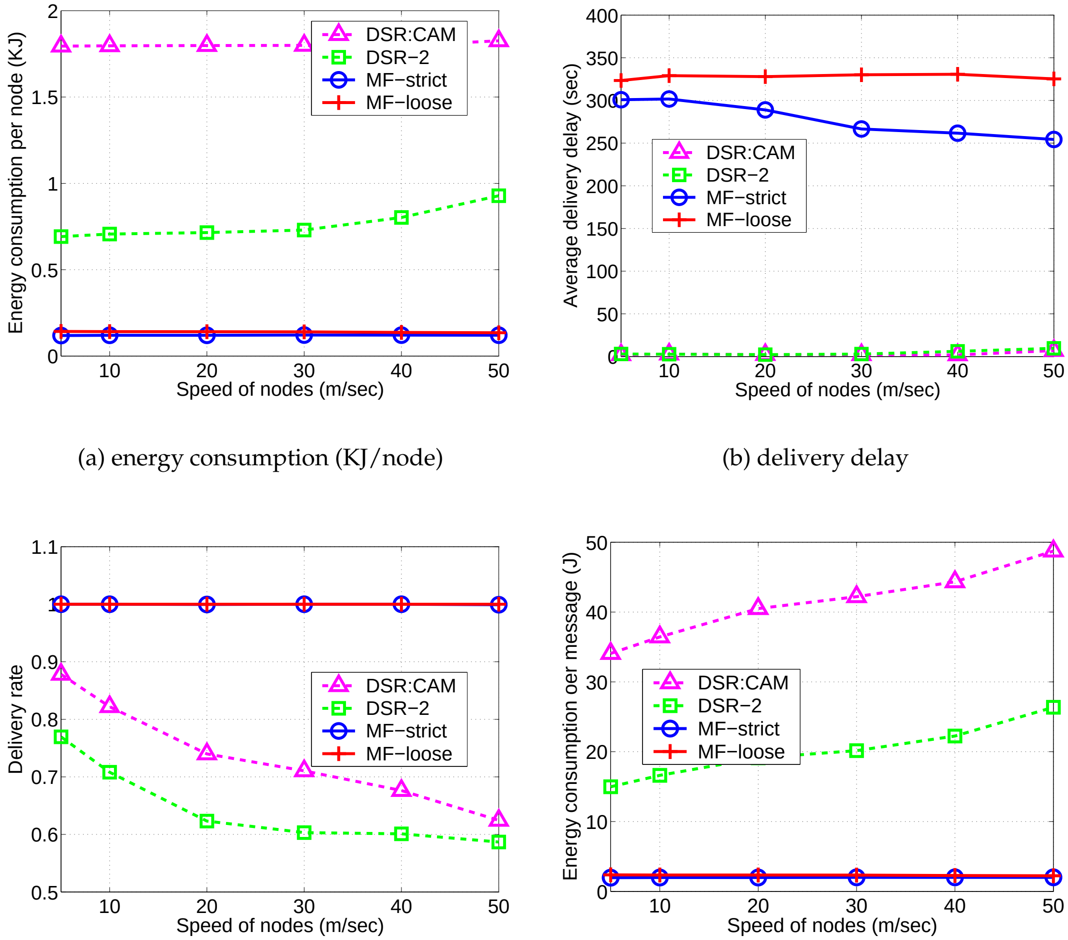
load is 300KB than when it is 3MB. Since the first message is dropped after a route change, the former loses a large proportion of messages to discover the route change than the latter. Thus, it has lower delivery rate. Beyond 3MB, the delivery rate decreases as the traffic load increases due to contention. In case of DSR-50 and DSR-200, both route change and contention decrease the delivery rate significantly. In MF, the node mobility decreases the length of communication time when a node meets the ferry. However, it increases the chance for a node to meet the ferry. Thus, the total communication time between a node and the ferry is not affected much by the node mobility, which results in the steady delivery rate.

For the loose schedule scenario, the pause time of a ferry causes time shift between expected meeting time and real meeting time between the ferry and nodes. We simulated with various mean pause times from zero to 420 seconds, which is the minimum time for a ferry to come back at the same location. However, the impact of pause time was too negligible to show as graphs. Therefore, we only summarize the results here. As the mean pause time increases, the ferry takes longer time to visit nodes. As a result, the delivery delay increases and the delivery rate decreases. Also, nodes spend more time in the searching mode instead of the sleeping mode because the real meeting time is much later than the expected meeting time, which increases energy consumption. The upper bound of this increase is the energy consumption that occurs when nodes are in the searching mode all the time except when they are in the communicating mode. This case occurs when nodes are mobile and the ferry has a loose schedule in our simulation. In such a scenario, our algorithm allows mobile nodes to sleep only when they become out of radio range from any point of the ferry route. Since the current ferry route covers the whole deployment area, a node would never become out of radio range from the ferry route and would never enter the sleeping mode. Thus, the energy consumption of this scenario provides the upper bound of energy consumption, and that is only slightly greater than that of DSR-50. In our simulation, the energy cost was also equivalent to that of DSR-50 when the traffic load is 3MBps.

6.4.3 Impact of Node Mobility

In this section, we evaluate the performance of MF and DSR as node speeds vary from 5 to 50m/s. This evaluation shows the robustness of the MF approach while the DSR approach suffers from network topology changes. Because the delivery rate of DSR-50 and DSR-200 is too low, we consider only DSR-2 and DSR:CAM.

Figure 44(a) shows the impact of node speed on the energy consumption. In MF, increasing node speed does not affect the energy consumption of nodes. However, it increases that in DSR because the high node speeds cause more route changes, obsoleting



(a) energy consumption (KJ/node)

(b) delivery delay

(c) delivery rate

(d) energy cost (J/message)

Figure 44: The impact of node speeds

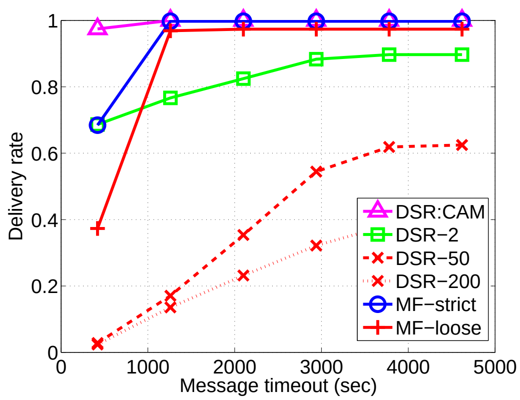
earlier routing information. To reconstruct the routing tables, DSR sends out route probing messages, consuming energy. In addition, if power management is used, nodes stay awake to forward the route probing messages. Due to the latter reason, the energy consumption of DSR-2 increases more than DSR:CAM.

Figure 44(b) shows the impact of node speed on the delivery delay. In MF, the delivery delay decreases as the speed of nodes increases because nodes meet the ferry faster as their speed increases. In DSR, increasing node speed increases the delivery delay of messages because route changes due to high mobility force nodes to probe routing paths again. Waiting for the results of route probing adds up to the delivery delay. However, it is minor compared with the delivery delay of MF.

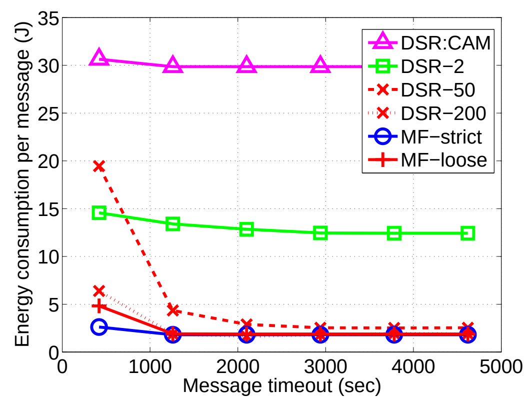
Figure 44(c) shows the delivery rate in DSR decreases as the speed of nodes increases because each route change causes the first message to be dropped while detecting the change. However, the node speed does not affect the delivery rate of MF.

Figure 44(d) shows that the energy cost of DSR increases as the speed of nodes increases because more messages are dropped at high speed scenario while more energy is consumed. However, that of MF does not change as the speed increases. Thus, DSR costs more energy to deliver a unit message when the node speeds are high.

6.4.4 Impact of Message Timeout



(a) delivery rate



(b) energy cost (J/message)

Figure 45: The impact of message timeout when nodes are stationary

In this section, we evaluate the performance of MF and DSR by varying the message timeout from 420 seconds to 4200 seconds. The timeout value of 420 seconds is a reasonable lower end for MF to be suitable because a ferry in our simulation setting usually takes more time to come back at the same location on its route. Because the message timeout enforces a maximum latency for each message delivery, this simulation provides the comparison between the MF and DSR approaches for a given fixed latency requirement.

Figure 45(a) shows that DSR:CAM delivers more than 95% of messages for all timeout values. However, when the timeout value is as small as 420 seconds, the delivery rate of MF drops to 68% in the strict schedule case and to 37% in the loose schedule case because the ferry may not meet destination nodes within the timeout. At the same time, the delivery rate of DSR with power management also drops because sleeping nodes forces their neighbors to store messages until they wake up to receive messages, which increases the delivery delay of messages. Also, power management reduces the time for nodes to exchange messages among themselves, which causes more contention and retransmission. The resulting retransmission increases the delivery delay of messages, which causes message drop due to timeout. On the other hand, Figure 45(b) shows that DSR:CAM has the highest energy cost. Therefore, DSR:CAM can deliver most of messages within given timeout values at the cost of energy.

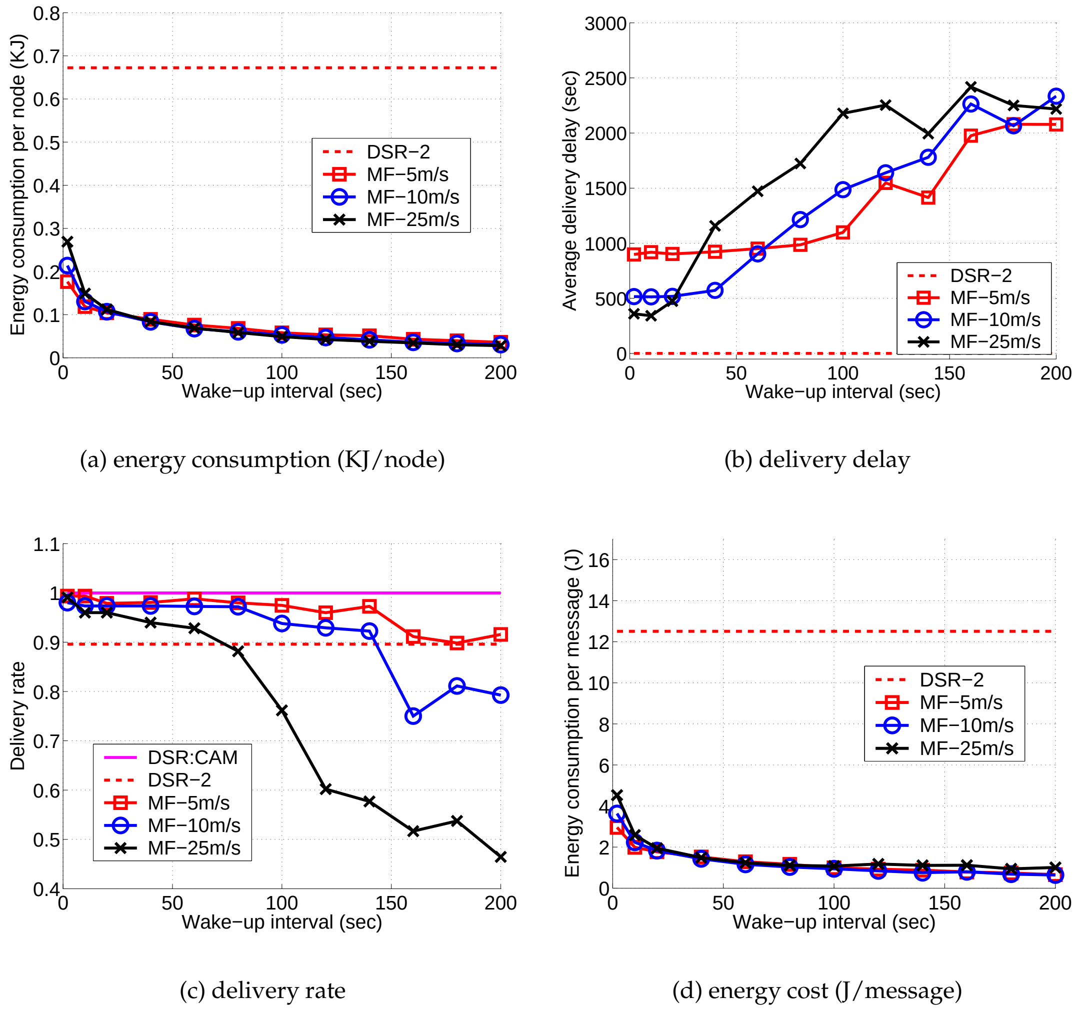
6.4.5 Impact of Wake-up Interval in Searching Mode

In this section, we evaluate the impact of wake-up interval in the searching mode of MF in a loose schedule scenario because the parameter trades between energy consumption and delivery delay. We vary the wake-up interval from two seconds to 200 seconds. We also vary the speed of the ferry as 5, 10, and 25m/s. In Figures 46 and 47, MF-x m/s represents the case of MF with power management where the speed of the ferry is x m/s, and its pause time at four corners of the route is exponentially distributed with an average of 50 seconds.

6.4.5.1 Stationary nodes

We first evaluate the performance of MF when nodes are stationary. Figure 46(a) shows the impact of the wake-up interval on the energy consumption. As the wake-up interval increases, the energy consumption of nodes decreases because nodes sleep more between wake-up events. In fact, the energy consumption is inversely proportional to the wake-up interval.

Figure 46(b) shows that the delivery delay increases as the wake-up interval increases, because a node may miss the ferry by waking up infrequently. If a node misses the ferry that passes through its radio range, the next chance comes when the ferry comes back. Figure 46(b) also shows that the speed of the ferry affects the delivery delay. When the wake-up interval is short, a faster ferry delivers messages faster because it moves faster.



(a) energy consumption (KJ/node)

(b) delivery delay

(c) delivery rate

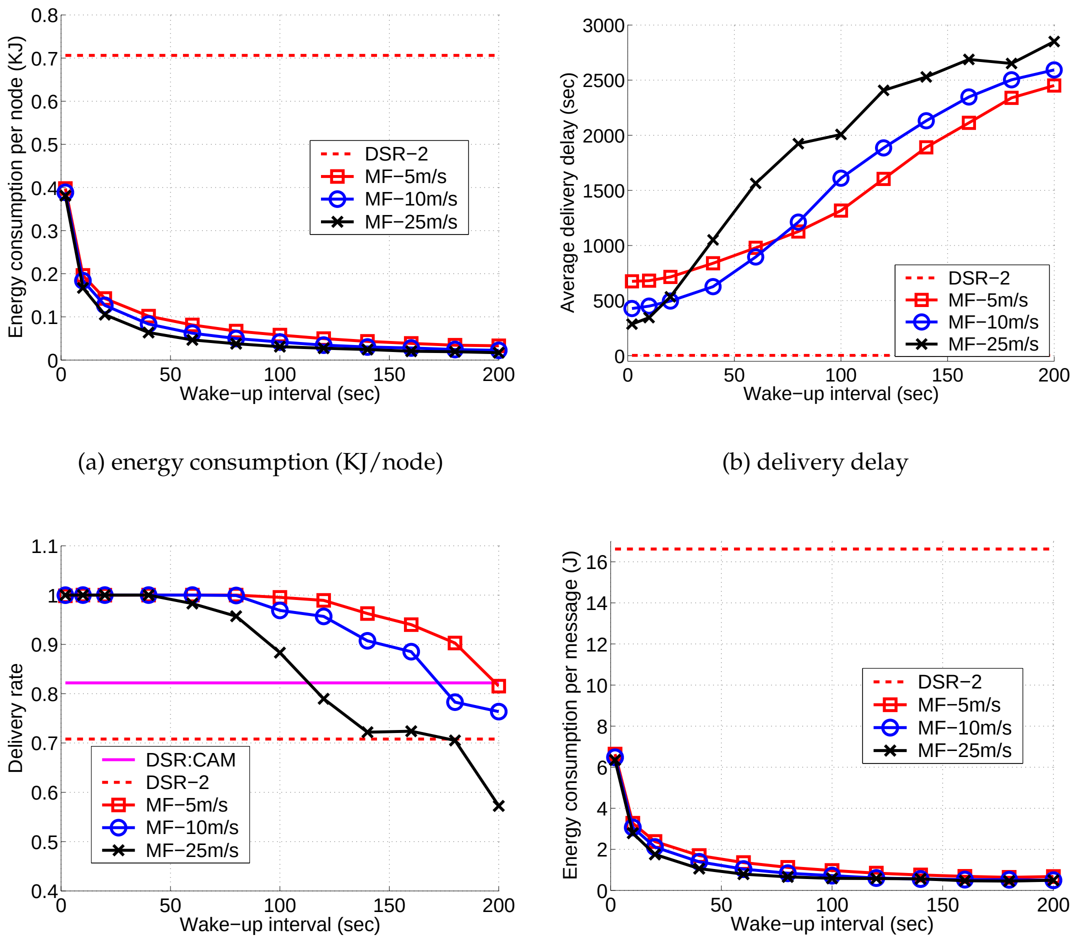
(d) energy cost (J/message)

Figure 46: The impact of wake-up intervals in the searching mode where nodes are stationary and the ferry moves on a loosely scheduled route

However, when the wake-up interval is longer than 20 seconds, a faster ferry stays in the radio range of a node for a short period of time, which increases the probability for a node to miss the ferry. As a result, the delivery delay of a faster ferry is longer than that of slower ferries when the wake-up interval is long.

Figure 46(c) shows the impact of the wake-up interval on the delivery rate. The delivery rate decreases as the wake-up interval increases, because a larger wake-up interval increases delivery delay. If messages are not delivered within 5000 seconds, they are dropped. Thus, a larger wake-up interval causes more message drops. Figure 46(c) also shows that the delivery rate of a faster ferry is lower than that of a slower ferry because more messages are dropped due to timeout. Finally, Figure 46(d) shows that the energy cost decreases as the wake-up interval increases due to overall energy savings.

6.4.5.2 Mobile Nodes



(a) energy consumption (KJ/node)

(b) delivery delay

(c) delivery rate

(d) energy cost (J/message)

Figure 47: The impact of wake-up intervals in the searching mode where nodes are mobile and the ferry moves on a loosely scheduled route.

We now evaluate the performance of MF when nodes are mobile. Figures 47(a) and (b) show the trade-off between energy and delay similarly to Figure 46. The degree of energy savings is greater than that of stationary nodes because mobile nodes spend more time in the searching mode than stationary nodes. Also, Figure 47(c) shows that MF has a better delivery rate than DSR under all three speeds of the ferry even when the wake-up interval is as large as 100 seconds. Because the radio range of the ferry swipes through the entire region over time, the ferry provides better network connection than DSR when nodes are mobile. Finally, Figure 47(d) shows that the energy cost decreases as the wake-up interval increases.

In summary, both stationary and mobile nodes can save energy by increasing wake-up interval in the searching mode. However, a very large wake-up interval reduces the delivery rate. Therefore, there is an optimal range of the wake-up interval in each network, and the wake-up interval that is less than 50 seconds is appropriate in these scenarios.

6.5 Summary

In this chapter, we investigate the use of the Message Ferrying routing scheme in a densely deployed network to save energy while trading off delay. We present a power management framework, in which nodes switch among different power management modes according to the knowledge of ferry location. Using ns-2 simulations, we evaluate the performance of MF and compare it with DSR. Our simulation results show that MF can achieve significant energy savings by trading off latency, while achieving high delivery rate. In contrast, power-managed DSR reduces energy consumption at the price of significantly lower delivery rate. In addition, MF shows robust performance in the face of node mobility. Therefore, delay tolerant applications can use MF rather than a multihop routing protocol to save energy when both routing approaches are available.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

In this thesis, we address many issues in power management for DTNs, including designing power management mechanisms and evaluating them for several mobility characteristics. We also provide systematic tuning parameters for network operators to balance between network performance and energy savings. In this chapter, we summarize the major contributions of this thesis and suggest several topics for future work.

7.1 Research Summary

A DTN is a generalized networking architecture to build network services in challenged environments with connection disruptions, where the disruptions occur due to many factors such as node mobility, physical obstacles, depleted energy, interference, and low node density. This DTN approach can be used for many important applications with energy constraints. Thus, efficient power management mechanisms are necessary to allow these networks to remain operational over a long period of time. However, mobile devices exhibit a tension between saving energy and providing connectivity. In order to pass messages, the device must discover other nodes, typically using the same wireless interface used for message delivery. At the same time, energy can be conserved by putting the wireless interface to sleep because the wireless interface is one of the largest energy consumers in mobile devices regardless of whether they are actively communicating or just listening. However, if the wireless interface is asleep to save energy, the node cannot communicate with other nodes. Thus, power management in DTNs must balance the discovery of other nodes while aggressively sleeping the wireless interface during the remaining periods. This thesis explores the issues in power management for DTNs and investigates on the following topics.

7.1.1 A Framework and Knowledge-based Mechanisms for a Single Radio Architecture

In this work, we investigate power management for a single radio architecture in DTNs. We present a power management framework, in which nodes switch among different power management modes according to the available knowledge about the network connectivities. In addition, we provide an explicit means to trade between energy and delivery performance when the statistical information about waiting time between contacts and contact duration is available. We also devise a mechanism to enhance the performance of our power management using the local traffic information in real time.

Our simulation results from various mobility scenarios show that the efficiency of the trading between energy and delivery performance depends on the network characteristics as well as the deployed mechanisms. Specifically, when partial knowledge is available, more knowledge about network dynamics (i.e., variance) is useful only when its value is relatively small compared to the mean value: in other words, when the node mobility is relatively regular. When the node mobility is random, our mechanism using only mean values discover contacts close to our desired ratio in MF. However, in RWP, it discovers a lot more than the desired ratio of contacts. Also, when the node mobility is intermediate, both mechanisms discover a lot more contacts than the desired ratio, consuming more energy than necessary. Therefore, it is worth to consider different power management approaches for networks with relatively random mobility. In the next work, we consider mechanisms in which radios search for contacts continuously without limiting its search within narrow time intervals in order to suit for the random mobility cases.

7.1.2 Hierarchical Power Management

In this work, we investigate power management in DTNs with high randomness in the node mobility. We present a hierarchical power management framework, in which nodes control two radio interfaces to discover contacts. Our simulation results from three mobility models show that our generalized power management mechanism balance between energy efficiency and contact discovery by tuning the wake-up intervals of the two radios.

Also, our experiment demonstrates that the wake-up intervals of the radios greatly impact energy efficiency and contact discovery performance, and the usefulness of the second radio is highly dependent on those intervals and mobility scenarios. More importantly, this does not answer the question as to how to set the interval in the first place. In the next work, we provide this as an analytical result when additional information about contacts and traffic load is available.

7.1.3 Traffic-Aware Optimization in Hierarchical Power Management

In this work, we devise traffic-aware approximation algorithms for hierarchical power management mechanisms, in which nodes control two radio interfaces to discover contacts. In our algorithms, nodes control wake-up intervals of its radios to save energy while discovering enough contacts to deliver the expected traffic load in the network. Our simulation results from three mobility models show that our generalized power management mechanism balances between energy efficiency and delivery performance by tuning the wake-up intervals of the two radios. In addition, when the traffic load can be estimated in advance, our approximation algorithms help nodes save significant amount of energy while handling the expected traffic load. Finally, the relative energy efficiency of using the additional low-power radio increases as the nodes' searching time increases.

So far, we have developed power management mechanisms in DTNs by passively observing the characteristics of DTNs. However, we came to ask a more fundamental question: Is the DTN approach only a back-up approach to resolve networking issues when traditional mechanisms do not work? In the next work, we investigate how to use a DTN approach proactively for energy savings even when the traditional approaches work.

7.1.4 Trading Latency for Energy Using Message Ferrying

In this work, we investigate the use of the Message Ferrying routing scheme in a densely deployed network to save energy while trading off delay. We present a power management framework, in which nodes switch among different power management modes according to the knowledge of ferry location. Using ns-2 simulations, we evaluate the performance of power managed MF and compare it with power managed DSR. Our simulation results show that MF can achieve significant energy savings by trading off latency, while achieving high delivery rate. In contrast, power-managed DSR reduces energy consumption at the price of significantly lower delivery rate. In addition, MF shows robust performance in the face of node mobility. Therefore, delay tolerant applications can use MF rather than a multihop routing protocol to save energy when both routing approaches are available.

7.2 *Future Directions*

This thesis has studied the issues of power management in DTNs. We suggest the following topics for the future work.

7.2.1 DTN Power Management Implementation

This thesis focuses on devising power management mechanisms for DTNs and uses simulations to evaluate them. The next step is to implement the proposed mechanisms. Usually, implementing such mechanisms introduces new challenges and new findings. For example, our power management mechanisms heavily depend on synchronized clocks among main radios as well as additional low-power radios. Since DTNs are partitioned, clock synchronization mechanisms proposed in wired networks or traditional MANETs will not work [50, 24, 27, 46, 52, 63]. Thus, in this thesis, we assume most of devices are equipped with GPS to synchronize their clocks [**?**]. Nevertheless, there are issues to be considered: i.e., how to make it work in the first place, how to trade between energy overhead to read GPS signals and accuracy of clocks, how to handle clock drifts among nodes, and what to do in the indoor scenarios in which GPS signals are not receivable. Also, each wireless card has limited configuration interfaces. Thus, it may not be feasible to implement the mechanisms exactly as proposed. Therefore, experience in the development of power management mechanisms would provide better understanding of power management in DTNs and guide the design of practical power management mechanisms in DTNs.

7.2.2 Adaptive Power Management in DTNs

This thesis explores stable networks in which nodes are mobile, but their mobility patterns are steady. Also, we assume the information about the network dynamics are collected in advance and a power management mechanism is run based on the information. However, DTNs occur in challenged environments, so the node mobility pattern or node deployment could change intentionally or unexpectedly. For example, a sensor network consisting of stationary sensors and mobile actuators could be scattered by an earth quake or a tornado. Therefore, the ability of detecting the changes and adapting the behavior such as routing and power management is important. For that purpose, in MaxProp, nodes records the likelihood to encounter other nodes whenever they meet other nodes and use the collected information to select routing paths [18]. However, it is questionable how to detect the change in a network while running a power management mechanism. In fact, when a power management mechanism is running, a node may miss contacts while sleeping even without the knowledge of missing them. A naive approach is to use a threshold to detect when the contact discovery ratio is far from the expectation. If it occurs, nodes could stay awake all the time to monitor the networks and collect the information, and then resume power management. However, this approach may waste significant amount of energy while monitoring the network. Thus, more sophisticated power management mechanisms that can save energy while updating the contact information would be beneficial.

7.2.3 Node Mobility Characteristics in DTNs

The key in designing routing protocols and power management mechanisms in DTNs is understanding node mobility characteristics. Analytical approaches as well as simulation-based approaches have been taken to understand the characteristics of node mobility and its impact on routing [13, 14, 33, 47, 49, 48]. This thesis utilizes the statistical information about time between contacts and contact duration to design power management. However, they may not be enough metrics to distinguish mobility characteristics. Thus, more systematic investigation would be interesting to understand the impact of node mobility characteristics on power management in DTNs.

7.2.4 Power Management for Opportunistic Routing in DTNs

The design of power management is greatly affected by routing protocols. In this thesis, we focus on knowledge-based routing, MF, and one opportunistic routing protocol (i.e., hot potato routing of the zero knowledge case in Chapter 3). In MF, nodes need to discover only the ferries, so they have chances to sleep when the ferries are far away. Therefore, the routing paths do not change. Also, in knowledge-based mechanisms, tuning parameters are determined by assuming that power management does not affect the routing paths as well as traffic load between nodes. However, in many opportunistic routing approaches [72, 69, 73], power management may affect routing paths and traffic loads among nodes because sleeping nodes may offload their traffic to awake nodes. Therefore, it would be interesting to design power management mechanisms for opportunistic routing as well as to explore their impact on routing and capacity in DTNs to balance between the network performance and energy savings in DTNs.

7.2.5 DTN Applications and Resource Management in Ubiquitous Computing

The technology evolution of mobile devices and sensors is leading us toward the ubiquitous computing era, in which computing systems are embedded into the environment. While the Internet and cellular phone systems depend on infrastructures heavily, it is not so clear how general mobile wireless networks in ubiquitous computing will look like in the future. However, most of mobile wireless devices have two problems: limited battery and radio interference. Because of the limited battery, people do not want to be on all the time. Also, if too many devices are equipped with high power, long range radios, they will interfere one another, lowering the system performance. Therefore, the resulting mobile wireless networks will have frequent connection disruption due to the intentional blackout or the short radio range. Thus, we believe the significant part of the future ubiquitous computing falls into the Disruption Tolerant Network (DTN) category.

It would be interesting to explore what applications DTNs can be applied for. Each application would require different performance guarantee. Thus, investigating the detailed requirements of each application and developing an optimal way to operate the system would be interesting. In addition, these systems operate with scarce resources such as limited battery. Also, they may have multiple choices of technologies to use with different costs such as WLAN, WiMAX, cellular phone systems, mobile ad-hoc networks, and sensor networks. Investigating the cost/value functions for different options would be beneficial to manage the scarce resource efficiently.

REFERENCES

[1] "Wireless LAN medium access control and physical layer specifications," August 1999. IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee).

[2] "NS-2 simulator." http://www.isi.edu/nsnam/ns/, November 2007.

[3] "RF/IF and Zegbee solutions." http://www.ti.com, November 2007.

[4] "UMassDieselNet project." http://prisms.cs.umass.edu/dome/, November 2007.

[5] ADYA, A., BAHL, P., PADHYE, J., WOLMAN, A., and ZHOU, L., "A multi-radio unification protocol for IEEE 802.11 wireless networks," in *IEEE Broadband Networks*, 2004.

[6] AKYILDIZ, I. F., ÖZGüR B. AKAN, CHEN, C., FANG, F., and SU, W., "InterPlaNetary internet: State-of-the-art and research challenges," *Elsevier Computer Networks*, vol. 43, pp. 75–112, 2003.

[7] AKYILDIZ, I. F. and KASIMOGLU, I. H., "Wireless sensor and actor networks: Research challenges," *Elsevier Ad Hoc Networks*, vol. 2, pp. 351–367, 2004.

[8] AKYILDIZ, I. F., POMPILI, D., and MELODIA, T., "Underwater acoustic sensor networks: Research challenges," *Elsevier Ad Hoc Networks*, vol. 3, pp. 257–279, 2005.

[9] AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., and CAYIRCI, E., "Wireless sensor networks: a survey," *Elsevier Computer Networks*, vol. 38, pp. 393–422, 2002.

[10] AKYILDIZ, I. F., WANG, X., and WANG, W., "Wireless mesh networks: A survey," *Elsevier Computer Networks*, vol. 47, pp. 445–487, 2005.

[11] ANAND, M., NIGHTINGALE, E. B., and FLINN, J., "Self-tuning wireless network power management," in *MobiCom*, 2003.

[12] BAHL, P., ADYA, A., PADHYE, J., and WOLMAN, A., "Reconsidering wireless systems with multiple radios," *ACM SIGCOMM Computer Communication Review*, 2004.

[13] BAI, F., SADAGOPAN, N., and HELMY, A., "IMPORTANT: A framework to systematically analyze the impact of mobility on performance of rouTing protocols for adhoc neTworks," in *IEEE INFOCOM*, 2003.

[14] BAI, F., SADAGOPAN, N., KRISHNAMACHARI, B., and HELMY, A., "Modeling path duration distributions in MANETs and their impact on reactive routing protocols," *IEEE Journal on Selected Areas in Communications*, vol. 22, September 2004.

[15] BANERJEE, N., SORBER, J., CORNER, M., ROLLINS, S., and GANESAN, D., "Triage: Balancing energy consumption and quality of service in a microserver," in *ACM/USENIX Conference on Mobile Systems, Applications, and Services (MobiSys)*, June 2007.

[16] BANERJEE, N., CORNER, M. D., and LEVINE, B. N., “An Energy-Efficient Architecture for DTN Throwboxes,” in *IEEE INFOCOM*, May 2007.

[17] BROCH, J., MALTZ, D. A., JOHNSON, D. B., HU, Y.-C., and JETCHEVA, J., “A performance comparison of multi-hop wireless ad hoc network routing protocols,” in *MobiCom*, 1998.

[18] BURGESS, J., GALLAGHER, B., JENSEN, D., and LEVINE, B. L., “MaxProp: Routing for vehicle-based disruption-tolerant networks,” in *IEEE INFOCOM*, 2006.

[19] CERF, V., BURLEIGH, S., HOOKE, A., TORGERSON, L., DURST, R., SCOTT, K., and TRAVIS, E., “Interplanetary internet (ipn): Architectural definition.” http://ipnsig.org/reports/memo-ipnrg-arch-00.pdf.

[20] CHEN, B., JAMIESON, K., BALAKRISHNAN, H., and MORRIS, R., “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” in *MobiCom*, 2001.

[21] CONNER, W. S., CHHABRA, J., YARVIS, M., and KRISHNAMURTHY, L., “Experimental evaluation of synchronization and topology control for in-building sensor network applications,” in *the Second ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2003.

[22] CULLER, D., ESTRIN, D., and SRIVASTAVA, M., “Overview of sensor networks,” *IEEE Computer*, 2004.

[23] DAVIS, J. A., FAGG, A. H., and LEVINE, B. N., “Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks,” in *Internatioinal Symposium on Wearable Computing*, October 2001.

[24] ELSON, J. and ESTRIN, D., “Time synchronization for wireless sensor networks,” in *International Parallel and Distributed Processing Symposium (IPDPS)*, 2001.

[25] FALL, K., “A delay-tolerant network architecture for challenged internets,” in *ACM SIGCOMM*, 2003.

[26] FEENEY, L. M. and NILSSON, M., “Investigating the energy consumption of a wireless network interface in an ad hoc network,” in *IEEE INFOCOM*, 2001.

[27] GIRIDHAR, A. and KUMAR, P. R., “Distributed clock synchronization over wireless networks: Algorithms and analysis,” in *Decision and Control*, 2006.

[28] GOLDENBERG, D. K., LIN, J., MORSE, A. S., ROSEN, B. E., and YANG, Y. R., “Towards mobility as a network control primitive,” in *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, (New York, NY, USA), pp. 163–174, ACM Press, 2004.

[29] HASSON, A. A., FLETCHER, R., and PENTLAND, A., “DakNet: A road to universal broadband connectivity.” Wireless Internet UN ICT Conference Case Study, 2003. http://www.medialabasia.org/.

[30] JAIN, S., FALL, K., and PATRA, R., “Routing in a delay tolerant network,” in *ACM SIGCOMM*, 2004.

[31] JAIN, S., DEMMER, M., PATRA, R., and FALL, K., "Using redundancy to cope with failures in a delay tolerant network," in *ACM SIGCOMM*, 2005.

[32] JAIN, S., SHAH, R., BRUNETTE, W., BORRIELLO, G., and ROY, S., "Exploiting mobility for energy efficient data collection in sensor networks," in *Modeling and Optimization in Mobile Ad Hoc and Wireless Networks (WiOpt)*, 2004.

[33] JIANG, S., HE, D., and RAO, J., "Prediction-based link availability estimation for mobile ad hoc networks," in *IEEE INFOCOM*, 2001.

[34] JOHNSON, D. B. and MALTZ, D. A., "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, 1996.

[35] JUANG, P., OKI, H., and WANG, Y., "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *ASPLOS-X*, (San Jose, CA), October 2002.

[36] JUN, H., AMMAR, M. H., CORNER, M., and ZEGURA, E. W., "Hierarchical power management in delay tolerant networks with traffic-aware optimization," in *ACM SIGCOMM Workshop on Challenged Networks (CHANTS)*, 2006.

[37] JUN, H., AMMAR, M. H., CORNER, M., and ZEGURA, E. W., "Hierarchical power management in delay tolerant networks with traffic-aware optimization," Tech. Rep. GT-CSS-2006-005, Georgia Institute of Technology, 2006.

[38] JUN, H., AMMAR, M. H., CORNER, M., and ZEGURA, E. W., "Hierarchical power management in delay tolerant networks using traffic-aware optimization," Tech. Rep. GT-CS-07-03, Georgia Institute of Technology, 2007.

[39] JUN, H., AMMAR, M. H., and ZEGURA, E. W., "Power management in delay tolerant networks: A framework and knowledge-based mechanisms," in *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, September 2005.

[40] JUN, H., AMMAR, M. H., and ZEGURA, E. W., "A framework and knowledge-based power management mechanisms in delay tolerant networks," Tech. Rep. GT-CS-07-08, Georgia Institute of Technology, 2007.

[41] JUN, H., ZHAO, W., AMMAR, M. H., ZEGURA, E. W., and LEE, C., "Trading off latency for energy in wireless ad hoc networks by message ferrying," Tech. Rep. GIT-CC-04-13, Georgia Institute of Technology, 2004.

[42] JUN, H., ZHAO, W., AMMAR, M. H., ZEGURA, E. W., and LEE, C., "Trading off latency for energy in wireless ad hoc networks by message ferrying," in *IEEE PerCom International Workshop on Pervasice Wireless Networking*, March 2005.

[43] JUN, H., ZHAO, W., AMMAR, M. H., ZEGURA, E. W., and LEE, C., "Trading off latency for energy in densely deployed wireless ad hoc networks by message ferrying," *Elsevier Ad Hoc Networks*, vol. 5, pp. 444–461, May 2007.

[44] KOMPELLA, R. R. and SNOEREN, A. C., "Practical lazy scheduling in sensor networks," in *SenSys*, 2003.

[45] KRAVETS, R. H., "Cooperative solutions to the dynamic management of communication resources," in *Ph.D. thesis*, 1998.

[46] LI, Q. and RUS, D., "Global clock synchronization in sensor networks," *IEEE Transactions on Computers archive*, vol. 55, pp. 214 – 226, February 2006.

[47] MCDONALD, A. B. and ZNATI, T. F., "A mobility-based framework for adaptive clustering in wireless ad networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, August 1999.

[48] MCDONALD, A. B. and ZNATI, T. F., "A path availability model for wireless ad-hoc networks," in *IEEE Wireless Communications and Networking Conference*, 1999.

[49] MCDONALD, A. B. and ZNATI, T. F., "Predicting node proximity in ad-hoc networks: A least overhead adaptive model for electing stable routes," in *MobiHoc*, 2000.

[50] MILLS, D., "NTP: Network time protocol." http://www.ntp.org/, November 2007.

[51] OTT, J. and KUTSCHER, D., "Drive-thru internet: IEEE 802.11b for "automobile" users," in *IEEE INFOCOM*, 2004.

[52] PALCHAUDHURI, S., SAHA, A. K., and JOHNSON, D. B., "Adaptive clock synchronization in sensor networks," in *the third international symposium on Information processing in sensor networks*, pp. 340 – 348, 2004.

[53] PARK, V. D. and CORSON, M. S., "Highly adaptive distributed routing algorithm for mobile wireless networks," in *IEEE INFOCOM*, 1997.

[54] PARTAN, J., KUROSE, J., and LEVINE, B. N., "A survey of practical issues in underwater networks," in *ACM International Workshop on UnderWater Networks (WUWNet)*, September 2006.

[55] PERING, T., RAGHUNATHAN, V., and WANT, R., "Exploiting radio hierarchies for power-efficient wireless device discovery and connection setup," in *VLSI Design*, 2005.

[56] PERKINS, C. E., "Ad-hoc on-demand distance vector routing," in *Presentation in MobiCom*, 1999.

[57] PERKINS, C. E. and BHAGWAT, P., "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM*, 1994.

[58] POLASTRE, J., HILL, J., and CULLER, D., "Versatile low power media access for wireless sensor networks," in *the Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2004.

[59] PRABHAKAR, B., BIYIKOGLU, E. U., and GAMAL, A. E., "Energy-efficient transmission over a wireless link via lazy packet scheduling," in *IEEE INFOCOM*, 2001.

[60] RAPPAPORT, T. S., ed., *Wireless Communications: Principles and Practice*. Prentice Hall, 2 ed., 2002.

[61] ROSS, S. M., *Introduction to Probability Models*. Academic Press, 2000.

[62] SHAH, R. C., ROY, S., JAIN, S., and BRUNETTE, W., "Data MULEs: Modeling a three-tier architecture for sparse sensor networks," in *IEEE Workshop on Sensor Network Protocols and Applications (SNPA)*, 2003.

[63] SHEU, J.-P., CHAO, C.-M., and SUN, C.-W., "A clock synchronization algorithm for multi-hop wireless ad hoc networks," in *Distributed Computing Systems*, pp. 574 – 581, 2004.

[64] SHIH, E., BAHL, P., and SINCLAIR, M. J., "Wake on wireless: An event driven energy saving strategy for battery operated devices," in *MobiCom*, 2002.

[65] SHURGERS, C., TSIATSIS, V., GANERIWAI, S., and SRIVASTAVA, M., "Topology management for sensor networks: Exploring latency and density," in *MobiHoc*, 2002.

[66] SINGH, S. and RAGHAVENDRA, C., "PAMAS: Power aware multi-access protocol with signalling for ad hoc networks," *ACM Computer Communication Review*, vol. 28, pp. 5–26, July 1998.

[67] SMALL, T. and HAAS, Z. J., "The shared wireless infostation model - a new ad hoc networking paradigm (or where there is a whale, there is a way)," in *MobiHoc*, 2003.

[68] SORBER, J., BANERJEE, N., CORNER, M. D., and ROLLINS, S., "Turducken: Hierarchical power management for mobile devices," in *MobiSys*, 2005.

[69] SPYROPOULOS, T., PSOUNIS, K., and RAGHAVENDRA, C. S., "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *ACM SIGCOMM Workshop on delay tolerant networking and related networks (WDTN)*, 2005.

[70] STEMM, M. and KATZ, R. H., "Measuring and reducing energy consumption of network interfaces in hand-held devices," *IEICE Transactions on Communications*, vol. E80-B(8), pp. 1125–1131, August 1997.

[71] TSENG, Y.-C., HSU, C.-S., and HSIEH, T.-Y., "Power-saving protocols for IEEE 802-11-based multi-hop ad hoc networks," in *IEEE INFOCOM*, 2002.

[72] VAHDAT, A. and BECKER, D., "Epidemic routing for partially-connected ad hoc networks," tech. rep., Duke University, 2000.

[73] WANG, Y., JAIN, S., MARTONOSI, M., and FALL, K., "Erasure coding based routing for opportunistic networks," in *ACM SIGCOMM Workshop on delay tolerant networking and related networks (WDTN)*, 2005.

[74] XU, Y., HEIDEMANN, J., and ESTRIN, D., "Geography-informed energy conservation for ad hoc routing," in *MobiCom*, 2001.

[75] YE, W., HEIDEMANN, J., and ESTRIN, D., "An energy-efficient MAC protocol for wireless sensor networks," in *IEEE INFOCOM*, 2002.

[76] YOON, J., LIU, M., and NOBLE, B., "Random waypoint considered harmful," in *IEEE INFOCOM*, 2003.

[77] YU, Y., KRISHNAMACHARI, B., and PRASANNA, V., "Energy-latency tradeoffs for data gathering in wireless sensor networks," in *IEEE INFOCOM*, 2004.

[78] ZHAO, W. and AMMAR, M. H., "Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks," in *FTDCS*, 2003.

[79] ZHAO, W., AMMAR, M. H., and ZEGURA, E., "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *MobiHoc*, 2004.

[80] ZHAO, W., AMMAR, M. H., and ZEGURA, E. W., "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *IEEE INFOCOM*, March 2005.

[81] ZHAO, W., CHEN, Y., AMMAR, M., CORNER, M. D., LEVINE, B. N., and ZEGURA, E., "Capacity enhancement using throwboxes in DTNs," in *IEEE Intl Conf on Mobile Ad hoc and Sensor Systems (MASS)*, 2006.

[82] ZHENG, R. and KRAVETS, R., "Asynchronous wakeup for ad hoc networks," in *MobiHoc*, 2003.

[83] ZHENG, R. and KRAVETS, R., "On-demand power management for ad hoc network," in *IEEE INFOCOM*, 2003.